# Evaluating relative importance of Pre-trained Encoders using Task Selection

**Anhad Mohananey**[1]
anhad@nyu.edu

**Pramit Mallick**[1]
pm2758@nyu.edu

**Srinidhi Goud**[1]
sgm400@nyu.edu

[1]**Department of Computer Science**
New York University

## Abstract

There has been considerable work that shows that multitask learning using hard or soft parameter sharing considerably improves performance. Many tasks such as Natural Language Inference(NLI) and Sentiment Analysis have been deemed suitable for pre-training encoder parameters. However, existing studies don't evaluate which tasks are suitable for using as encoders for certain target tasks. The systems presented in existing literature are also less flexible for adding new tasks. We propose a method that employs a neural network to select a sentence representation from multiple encoders, each trained on separate tasks. Such a setting is hard to train, due to the fact that making a discrete encoder task selection decision makes the model non-differentiable. We propose a method that addresses this issue using the gumbel softmax(Jang et al., 2016), wherein a discrete decision is made on the forward pass, but the function being continuous, is possible to train using gradients and back-propagation. We run experiments to study the relative importance of pre-trained tasks for a certain target task.

## 1 Introduction

Word embeddings (Mikolov, 2013) have been shown useful for improving performance of neural network models. Sentence level representations are a more recent research area, where in the entire sentence is represented by a fixed length vector. Various supervised learning objectives can be used to train these sentence representations. Some of them are Machine Translation, Natural Language Inference (Bowman et al., 2015), Sentiment Analysis (Socher et al., 2013) and Duplicate Questions checking. Soft parameter sharing in Multitask learning (Ruder, 2017) involves training multiple models trained on separate tasks. Recent work in multitask learning has shown that sentence representations trained using a particular task can be employed to boost performance for another task. However, usually in these settings, the neural models consist of multiple encoders or a shared encoder. The inherent problem with these models is that is that they are not easy to generalize. In case a new encoder is added, the entire structure needs to be trained again. Another issue is that they give us very little information about which pre-trained task encoder is most useful for the current task at hand. In other words, it is difficult to make a judgment about task hierarchies. This work introduces a task selection method that helps access the relative importance of pre-trained tasks for the current target task. In other words, we move towards the goal of building general task hierarchies.

We propose a multi encoder model in which there are multiple encoders, each pre-trained on different tasks. During inference time, only one of these encoders is used for feeding into the classification model or decoder. The decision of choosing the encoder is made by a neural network based classifier, that is trained in conjunction with the model. A discrete decision made about which sentence encoder to use makes this model non-differentiable using standard techniques. To solve this problem, although we perform discrete decision, we use the Straight Through Gumbel Softmax(Bengio and Courville, 2013), that makes it possible to compute an approximate gradient to make back propagation work. Due to the fact that our model selects only one particular task for encoding, it is possible to draw judgements as to which task is more important as a pre-trainer for the target task. Also, this model is more flexible. If a new encoder is added from a separate task, the neural network can be trained in a way to prefer the older tasks till the weights are learned properly.

This paper evaluates whether this approach based on task selection gives any significant performance boost over using just a single pre-trained task. The primary goal of this study is to provide a detailed analysis based on experimental data, as to which tasks are better for certain downstream tasks.

## 2 Background Motivation

Multi-task learning (MTL) attempts to increase generalization and performance by borrowing domain-specific information and representation contained in the training of related tasks. While that concept easily translates in vision tasks due to the nature of signal, multi-task learning in natural language proves to be trickier due to the discrete nature of the data. Typically, this approach is used when there is a need to learn representation(s) that can be used across various tasks and types of tasks while maintaining semantic sense. (Ruder, 2017) describes two approaches to MTL mainly differing in the way the parameters of the models are shared.

Soft parameter sharing is one where different models (or encoders) are trained separately on different tasks but a regularization is added on the parameters to encourage the hidden representations to be similar and to borrow information from each other (between tasks). (Duong et al., 2015) exploits the underlying shared structures across languages and information using (Chen and Manning, 2014)'s parser's help to learn dependency parsing of a target language (one which has little supervised data/resources available). (Yang and Hospedales, 2016) tries a similar technique albeit applied to the vision domain.

Hard parameter sharing is the most commonly used technique which shares part(s) of models across the supervised learning of tasks. This ideally reduces the risks of overfitting. Seq-seq models attempting translation tasks (English-to-Spanish-to-French-to-German) naturally extend this need and are typical use cases for this approach. (Luong et al., 2015) explores various settings, one-to-many - where they train a single encoder but multiple decoders (useful for multi-target translation or to learn a common representation), many-to-one - multiple encoders to a single decoder (useful when the target task is a combination of source tasks such as German image captioning). (Søgaard and Goldberg, 2016) experiments with MTL in the context of hard parameter sharing for sequence tagging using layered bi-RNNs where the network at each layer is trained on a different task (allowing for a enforced hierarchy of tasks and cascaded learning). Their experiments and result confirmed the intuitive idea that lowlevel tasks are better kept at the lower layers, enabling the higher-level tasks to make use of the shared representation of the lower-level tasks, thus reinforcing the idea of existence of a hierarchy between the different tasks.

An interesting exploration by (Mou et al., 2016) reveals that the effectiveness of transference of neural networks, whether it be through parameter sharing or fine tuning of pre-trained weights or training in a cascaded manner or using a joint objective function, depends on how closely the concerned tasks are semantically related. Other insights augment previous observations about the final layers of the model being specific to the final task, but the lower layers (such as word embeddings) being transferable. The semantic relatedness also lends to the idea of task-relations. (Bingel and Søgaard, 2017) study this very claim in the context of sequence labelling. Using several tasks such as CCG, chunking, POS, compression, as main or auxiliary tasks, they analyze gains in MTL by studying the learning curves of the tasks and reveal an insightful observation - MTL gains are more likely for target tasks that quickly plateau with non-plateauing auxiliary tasks. However, it is not easy to identify the best auxiliary task for the task at hand and the work does not extend to a hierarchy of tasks for learning. Another investigation was done (Conneau et al., 2017), who recognized the Natural Language Inference task [(Bowman et al., 2015)] as being the most downstream task, supervised learning on which leads to learning of universal sentence representation. However the investigation falls short on experimenting with larger corpuses and varied tasks and the possibility of a new task.

Our investigation is motivated by the need for understanding MTL better and to identify which task relations matter/are important. We propose an umbrella technique for understanding MTL using pre-trained (on different tasks) models and using the novel Gumbel Softmax (Jang et al., 2016) function which gives a categorical decision during the inference phase at the same time allowing back-prop through a differentiable function.

## 3 Models and Methods [1]

This paper defines a methodology wherein an encoder is chosen from a set of N possible encoders, to be applied to a standard neural network classification architecture. Each of the N encoders is trained separately on different tasks. The decision of selecting which encoder representation to use is done using the Straight Through Gumbel Softmax function. Since there is no possible supervised signal for this decision network, it has to be trained using the final task objective. Traditional Softmax with discrete decisions wouldn't work in this case because the model would be come non-differentiable, thus making it impossible to approximate gradients.

Gumbel Softmax(Jang et al., 2016) is a method of utilizing discrete random variables in a network. It approximates one-hot vectors sampled from a distribution by making them continuous. Approximation as continuous makes the gradients calculable using standard back propagation and the reparameterization trick. Given unnormalized probabilities $k^1, k^2..k^n$, a sample $y^i$ from the Gumbel distribution is given by:

$$y^i = \frac{exp((log(k^i) + g^i)/temp)}{\sum exp((log(k^i) + g^i)/temp)} \quad (1)$$

$$g^i = -log(-log(u^i)) \quad (2)$$

$$u^i = Uniform(0, 1) \quad (3)$$

In the above equation, $g^i$ is the gumbel noise, and $temp$ is the temperature parameter. As $temp$ becomes closer to 0, the distribution becomes similar to a one-hot encoding. The Straight Through Gumbel(Bengio and Courville, 2013) takes different paths for the forward and backward passes. During the forward pass, it takes the argmax of $y^i$ values. On the backward pass, it uses the continuous representation of the $y$ values, thus back propagation is possible.

Each encoder is composed of a sequential LSTM trained individually on the particular task objective. Consider the case of three LSTM encoders $Task1$, $Task2$ and $Task3$. Here each of $Task1$, $Task2$ and $Task3$ have been trained on a different task/objective function separately. A linear layer takes outputs from the three LSTMs, and runs the Straight Through Gumbel estimator on top of it. The estimator acts like a task selector, making a discrete decision on which of the three representations to pick for the layers ahead. The whole model is trained based on the final task supervised objective.

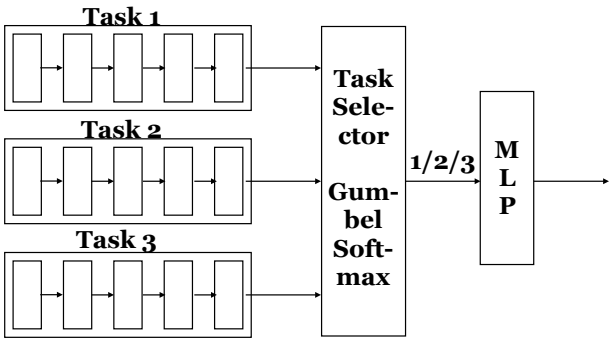


Figure 1: Model Architecture

### Tasks, Dataset and Additional Details

We start off by experimenting on three tasks - natural language inference (Bowman et al., 2015), sentiment classification (Socher et al., 2013) and the Quora duplicate questions task [2] to demonstrate a proof of concept. We plan to explore more with Neural Machine Translation(NMT) and other complex tasks later. All models are implemented in PyTorch, taking inspiration from the NYU-SPINN code base [3] while adding the necessary changes of our own. Our code, saved models, and output log files are available on GitHub.[4]. As is standard, we use the 300D 840B GloVe vectors (Pennington et al., 2014) for word embeddings and pad all the sentences with zeros to match the largest sentence. This helps us design a more elegant and efficient batcher for the sequence models. These sequences are fed to a bi-directional LSTM that again produces a 300D sentence embedding. This forms the encoder part of the model that we use later during the task selection process. The final layer of the pre-training process is an MLP layer that trains the task specific model. We choose the bi-directional LSTM framework as the encoder model as it is believed to be largely successful in most sequence learning tasks (including NMT (Bahdanau et al., 2014)). We keep a small

[1] https://github.com/pramitmallick/NLU-Project

[2] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[3] https://github.com/nyu-mll/spinn

[4] https://github.com/pramitmallick/NLU-Project

learning rate of 0.001 and a momentum of 0.9 with an L2 regularization coefficient of 1e-5 using the Adam optimizer while also adding a dropout of 0.1 to the encoders and classifiers to help better generalize. We train all the tasks over 10 epochs. Emphasis is not made in tuning these hyperparameters as we feel that getting the Gumbel layer and task selection experiments to work properly would be a more judicious use of our time. For the final task selection process, we are in the stages of coding out the required networks and models.

## 4 Contributions

While we all helped each other out and contributed to all the sections via code review and proof reading. However, the coarse level contributions are -
Anhad - Coded out the model for SNLI and carried out the relevant experiments. Wrote the introduction and abstract section of the draft.
Pramit - Coded out the model for Quora Duplicate Dataset and carried out the relevant experiments. Wrote the background and motivation section of the draft.
Srinidhi - Coded out the model for SST and carried out the relevant experiments. Wrote the models and methods section of the draft.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Yoshua Nicholas Lonard Bengio and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* .

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303* .

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* .

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 740–750.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* .

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 845–850.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* .

Tomas et al Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111* .

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* .

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 231–235.

Yongxin Yang and Timothy M Hospedales. 2016. Trace norm regularised deep multi-task learning. *arXiv preprint arXiv:1606.04038* .