Final Project Report: Semantically-related Data Ordering for Neural Network Models

Natalia Wojarnik, Cutter Dalton

November 2022

1 Abstract

Recent advancements in Artificial Intelligence have been predicated upon using massive, complex models. While these models may perform markedly better than their more simplistic counterparts, they require a tremendous amount of training time and processing power. Thus, there is a demand for alternative approaches to improving model performance. Inspired by curriculum learning, this paper demonstrates the efficacy of semantic-related data ordering, an approach which involves intentionally ordering a model's training data to improve learning. We conducted experiments involving topic classification using the News Category Dataset on two separate models; one predicated on a simple RNN, and another which incorporates bidirectional LSTMs and a GRU. We report significant improvement in performance for the models which use ordered datasets, and peak performance from datasets which emulate curriculum learning.

2 Introduction

2.1 Motivation

Recent years in AI showed a significant improvement in the development of complex and robust models being able to solve multiple NLP or image related tasks. It pushed the field towards creating huge open-source datasets available for the purpose of training smaller neural networks. As the growth of model sizes, its parameters and consequently the need for computer power, does not stop, there is a great importance put on the datasets fed into the model. Focus in recent years shifted to the discussion about data quality, data biases, data security or data augmentation when needed. We suggest a focus on data ordering based on the semantic-level schemes to check if the model's performance and speed of convergence improves. Instead of building bigger models, we make use of the already existing available data to make up for the lack of computation power and scalability, but produce similar results: better performance, faster learning.

2.2 Existing Solutions

The method that is gaining popularity in machine learning is curriculum learning. The idea we present is similar to this approach. Curriculum learning describes a learning approach in which the model is fed with easy examples first and then gradually, as the model learns, the task difficulty increases. The idea comes from the way humans gain knowledge. As in kids' development, the model at the beginning incrementally learns easier tasks and later, as humans become adults and gain experience, the model is trained on the examples that appear to be more difficult. Research already shows better performance of the models where the training data is ordered by its perplexity: from the training examples that have the highest accuracy to the ones with the lowest accuracy at the prediction level (source). These experiments were conducted on image recognition models and applied to image datasets. Moreover, for machine translation tasks, the models were fed with the words from training sentences from target and source languages ordered by their dependency structures. Other experiments with curriculum learning were conducted using a voice recognition model with increasing number of speakers and the level of data augmentation, or with a multi-modal language model ordering language data from concrete to abstract concepts. This indicates that the idea of data ordering has been used before. However, none of the existing approaches take simple semantic categories into consideration, and none of them order the training data using its labels to group them into semantically similar clusters. Most of the ideas involving curriculum learning, as well as the curriculum learning as an approach in itself, assume that ordering the training data is done followed by the experiments on accuracy/perplexity of the model after seeing the training data for the first time. We propose not inducing any model influence before ordering the data but mimicking the semantic similarity of the training data known from human experience. The references below mention the problem arising in training neural networks which is catastrophic forgetting. It is a tendency of a model to completely and abruptly forget previously learned information upon learning new information. This may be a difficulty in training models where data is grouped semantically and the categories that come first are forgotten before the next categories occur in the training data. However, as indicated in the papers mentioned below, data ordering was used to overcome catastrophic forgetting phenomena and our proposed semantic categorization with the weights adjustment can be a solution for the model to memorize crucial information about the data. This tendency is worth mentioning and exploring with regard to data ordering since none of the papers suggest semantic ordering as a solution to the catastrophic forgetting problem, as well as online learning experiments do not take this ordering into consideration when increasing incrementally the size of training data.

2.3 Improved Solutions

Inspired by the way humans memorize sequences of items, we propose combining the training data examples into semantically similar groups. Eg. humans will learn a sequence of numbers easier when the values of numbers close to each other share a pattern (3,4,5,90,80,70,5,10,15) or a sequence of a grocery list when the items next to each other are semantically similar ([fruit] apples, oranges, peaches, [cleaning supplies] cloth, laundry detergent, broom). Similarly, the training data ordered into semantic groups may produce better results in terms of performance and faster model convergence, therefore faster learning phase. For image recognition or object detection models, the examples picturing only one semantic group are next to each other while training. The semantic groups include different layers of granularity and can be broken down into more fine-grained less abstract semantic categories. Eg. images of animals would be in the first part of all training examples and within this one semantic group other animal categories will be broken down and their examples will be ordered to appear next to each other (all the images of dogs followed by all the images of cats, etc). For text datasets the ordering depends on the NLP task and dataset structure. The texts already labeled for genre or topic would be grouped together by their semantic groups, eg. news first, followed by short stories, followed by social media posts or for news text dataset: war, government, politics, education, health, the environment, economy, business, fashion, entertainment, and sport, etc. Another approach would involve topic modeling first and then clustering the texts in training data based on their semantic similarity putting examples from one topic next to each other. We are not sure yet how many modalities we can take into consideration within the scope of the project and how complex mechanisms we can use to create the similarity groups besides the simple data ordering based on labels. It would be interesting to see if the audio data ordered by its similarity produces also better results while ordered eg. within one music genre or one instrument type. We would like to focus on image and text datasets first with possible extensions. By applying a more general way to extract semantic categories, we would be able to conduct experiments on muli-modal tasks. The experiments we would like to conduct will focus on the impact of semantic data ordering on model performance, speed of convergence and possible solutions to catastrophic forgetting (grasping and remembering the meaning of semantic groups) based on data ordering.

3 Related Work

There exist three main different domains in the field of neural network models that entail the idea of data ordering and that we will focus on: data ordering in curriculum learning approach (online learning), data ordering for sequence processing in neural network models (catastrophic forgetting) and adversarial data ordering for the purpose of manipulating and slowing down the model performance (data attacks).

3.1 Data Ordering and Curriculum / Online Learning

The related work in data ordering that has been already proposed is mostly centered around the idea of Curriculum Learning, first introduced by Elman in 1993 [4]. The concept bears on the assumption that the model learns better and faster when trained on a data set ordered from easier instances to more difficult examples. Most of the experiments done in the field take into consideration the perplexity score which, after the model starts training, tells us how much trouble the model has with predicting correct answers. The paper from 2018 [13] introduced an innovative idea of Stochastic Curriculum Learning (SCL) which entails adding more data with the next batches coming in the training based on the difficulty score. Difficulty level of the examples is measured not by human judgment but by another classifier. The CIFAR dataset used for conducting their experiments is broken down into different levels of granularity for image classification. More fine-grained categories, described as super-classes in CIFAR, were treated as more difficult training examples. Classes and objects, so examples categorized as more broad in CIFAR dataset were fed to the model in early stages of training. The final model achieved better performance while utilizing curriculum learning in the training phase. The curriculum learning approach expanded vastly in recent years and is being utilized in several NLP tasks, such as Machine Translation and Neural Machine Translation. As proposed first in 2014 [2], the idea of data ordering led to successful results and better final results on the Chinese-English machine translation model. The experiments of work from 2014 were based on the word order in sentences. The position of certain words were changed according to the dependency parsing rules of the sentence to match the linguistic underlying structure. The proposed idea has strictly linguistic ground and does not involve changing the order of the training examples in itself, but the order of tokens in a sentence sequence. Later work on the Neural Machine Translation model from 2019 [5] expanded significantly on the idea of data ordering for this NLP task. Researchers conducted a suite of empirical studies and experimented with various ordering strategies. Their approach differs from the online learning concept such that the network can access all the training samples at once, not incrementally in batches which is the case of online learning. The training data is reordered once and then, the model makes the predictions based on these ordered instances. Every training example includes a pair of sentences, in a source language and a target language. The work presents 4 different strategies: randomly shuffling the data; sequence length ordering based on the length of sentences (in the ascending and descending order); perplexity based ordering (in the ascending and descending order); and BLEU score based ordering (in the ascending and descending order). The experiments from the paper conclude that the strategy of ordering the data by the perplexity score in the ascending order improves BLUE score on the final model of 1.7. Easy sentence pairs at the beginning of the training and increasing difficulty of the sentence pairs to translate seem to work well with the promising results. Another NLP experiment done with curriculum learning and data ordering was presented in the paper from 2021 [10]. The authors combined the multi-model representations, textual from BERT and visual from ResNet-152 and constructed a new dataset from Wikimedia Commons based on the categorization: concrete/abstract words. Certain words were first categories as less or more concrete according to the existing scale (0-7) and the images that were captioned with these words were ordered: from the most concrete to the most abstract. The idea relies on the curriculum learning premise that the most concrete objects are the easiest to learn with the increasing difficulty when dealing with more abstract concepts and their image representations respectively.

Another work from 2021 [3] suggested adopting the curriculum learning and data ordering to help improve neural data-to-text generation. The final model trained with curriculum learning principles increased performance. The authors experimented with various difficulty metrics, such as length of sequence in text data assuming that the longer the sentence, the more difficult it is for the model and should be placed later in the training. Another metric was word rarity. The more rare words in a sentence, the harder it is for the model to learn. Additionally, the author of the paper introduced a new method of difficulty measurement - “soft edit distance metrics”, calculated based on the number of operations needed to be completed by the model: delete, insert, edit. The final model supported by curriculum learning has faster convergence speed where training time was reduced by 38.7% and performance was boosted by 4.84 in BLEU score.

Image classification is another task that shows better performance while using curriculum learning. The work from 2022 [8] proposed five different methods of data ordering. Two methods involved ordering the data a priori (with human-in-the-loop approach). Other three methods included ordering rules specified after the model started learning. Ordering was done specifically to detect misclassified examples in the training data and use these instances as the ones that are more difficult to learn for the model. An ordering based on the embeddings produced by convolutional neural networks pre-trained on ImageNet had the best final performance which concludes that data ordering accruing to curriculum learning principles increases results.

One of the most recent papers from 2022 presented the speech verification experiments with data ordering in the curriculum learning fashion [7]. Authors improved the final model performance with the 30% increase over the baseline. The volume of training data was gradually added with the increased number of speakers. Another data ordering mechanism was data augmentation. The training samples were gradually being added in mini-batches with increasing noise and reverberation. The premise of the experiment was that the more noisy data, and parallelly, the more speakers in the conversation, the more difficult data gets. Therefore, these harder examples were ordered as later in the training set. The model used in these experiments is a self supervised vision transformer DINO framework for speaker verification.

As proved by multiple sound tasks, a data ordering approach brings successful results and boosts AI model performances across different domains. However, as shown in the above examples, data is exclusively ordered in the curriculum fashion and the ordering strategies are limited to the purpose of categorizing training instances from the easiest to the harvest for the model to learn. Several experiments used classifier supported approaches to categorize the level of difficulty, some proposed human judgment based classification schema. There were attempts to semantically cluster the training data according to its concreteness saturation but again, the premise of data ordering was rooted in the understanding that the model will converge faster and learn better when difficult examples are followed by easy examples, from the most concrete to the most abstract.

Our proposed data ordering strategy is different from all the above approaches in the way that it takes into consideration only the semantic clustering, i.e. grouping the data examples that are labeled with the same gold standards. Semantic clustering in our work refers exclusively to the way humans understand semantics. We understand that two inputs can be semantically related in almost an infinite number of ways and the experiments described above touched on this topic as well looking for new different ways to map the data examples into category buckets, such as NLP experiments with word rarity levels, sequence length groups and linguistic difficulty tiers. By designing a suite of experiments with semantic clustering data ordering, we would like to investigate orderings that can be congruent and incongruent with the learning task. For example, an image classification model could predict animate/inanimate at the output when a white car and a white dog would be semantically dissimilar, but for prediction tasks requiring outputting the color, these two inputs would be semantically similar. The way we will approach data ordering experiments comes first from the exclusively human inspired embodiment theory of how we learn by concepts and grouping them into semantic clusters. Then, we will attempt to compare these human-inspired ordering strategies to the one that are inspired by what we know about the way models learn. We do not try to approach the data ordering from the curriculum learning perspective exclusively. Some of the data orderings might follow the curriculum learning fashion but it's not the foundational premise of our experiments.

[5] [2]

3.2 Data Ordering and Sequence Processing / Catastrophic Forgetting

Data ordering, besides the curriculum learning approach, is broadly used in sequence to sequence models like LSTM to help the models memorize sequential data better and be able to use data ablation strategies to decrease the need for model scalability. As mentioned in the 2022 paper [11] data ordering was a successful approach in neural machine translation to alleviate the imbalance training problem by introducing the proposed Complementary Online Knowledge Distillation (COKD), which uses dynamically updated teacher models trained on specific data orderings to iteratively provide complementary knowledge to the student models. Another work from 2020 [6] proved that data ordering might be beneficial for "memorizing" information for neural network models. The authors proposed REMIND, a streaming learning model that learns what to remember from the weights and what to forget by ordering the training data examples. The REMIND model achieved state of the art results for object classification tasks and outperformed other methods for incremental class learning on the ImageNet trained model. Even though the described issues are being solved with a data ordering paradigm, we do not plan to focus on the catastrophic forgetting and imbalanced data problems. The results of our experiments may serve as the foundations for further exploration in the domains of network ablation, catastrophic forgetting or network dissection but it is not the leading motivation.

[13] [4] [7] [10] [3]

3.3 Data Ordering and Adversarial Data Attacks

Data ordering has been explored as a mechanism for slowing down and preventing the model from learning. This domain focuses on the negative purposes of using data ordering. As the work from 2021 suggests, the training sets can be ordered in an adversarial way and serve as the danger of data manipulation not letting the model learn or leading it to learn in a certain direction. As shown, with the use of stochastic gradient descent, data ordering techniques can also be utilized in a negative way being a danger for attacks. However, we do not intend to investigate this idea in the presented work. Ethics in AI is certainly of great importance, however the mentioned papers showed, data fed into models can be very easily manipulated and misused, for example by ordering data in a desired way. By using the semantic groups data ordering, we will try to argue that data ordering can bring

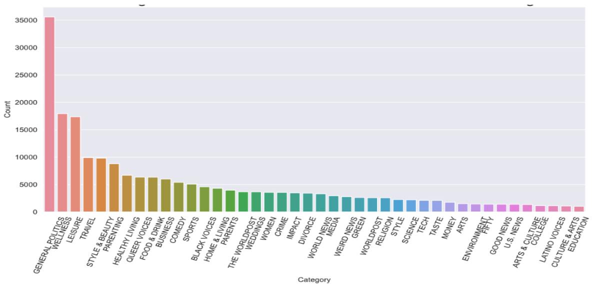


Figure 1: Distribution of the 42 Classes

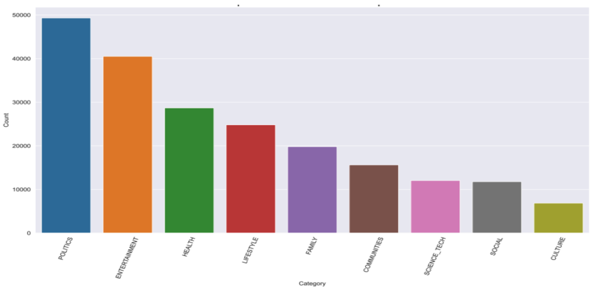


Figure 2: Distribution of the 9 Superclasses

beneficial and objective results while capturing the semantic understanding of a certain category. [8] [9] [11] [6] [12] [1]

4 Methods

In order to evaluate the efficacy of human-inspired semantic clustering with machine learning, we will be conducting experiments on the text data.

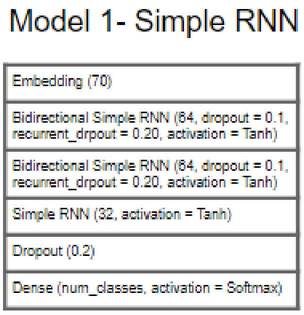
4.1 Data

The News Category Dataset used to conduct the experiment is available open-source on Kaggle https://www.kaggle.com/datasets/rmisra/news-category-dataset. This dataset contains around 210k news headlines from 2021 to 2022 from HuffPost. This is one of the biggest news datasets and served as a benchmark for the category prediction. The baseline mentioned in the experimental results refers to the predictions of news categories based on this dataset. Baseline of the experiments means that the training data was shuffled after splitting into training and testing batches in the regular form using the Keras methods. The baseline results are compared to the experimental setting where the training examples were purposefully ordered in a certain way. More about the experimental design in the next section. There are a total of 42 news categories in the dataset which are referred to as classes. The 42 classes were grouped by the authors into 9 superclasses and the predictions were made on both: classes (more fine-grained) and superclasses (less fine-grained). The classes and superclasses are shown on the figures 1 and 2.

4.2 Models

The experiments were run on 2 different models whose architecture is shown on the figure (PUT FIGURES). The first model is a simple RNN model. It starts with an Embedding layer of size 70 and is followed by 2 layers of Bidirectional Simple RNN of size 64 with dropout 0.1 setting, recurrent dropout 0.2 and activation function tanh. Next layer of the network is Simple RNN of size 32 with the activation function tanh. The mode 1 uses Dropout 0.2 and the last layer constitutes the Dense layer for making predictions. The size of the last layer depends on the number of labels predicted (for classes - 42, for superclasses - 9). The activation function used in the last layer is softmax to enable the multi-label prediction of the model.

The model 2 used for the experiments has a bigger architecture including LSTM and GRU, therefore was significantly slower. The model starts with the Embedding layer of size 100 and is followed by 1 layer Bidirectional Simple LSTM of size 64 with dropout 0.1, recurrent dropout 0.1 and activation function tanh, followed by 1 layer of Bidirectional Simple LSTM of size 64 with dropout 0.1, recurrent dropout 0.1 and activation function tanh. Next, the model includes a layer of Bidirectional Simple RNN of size 64 with dropout 0.2 setting, recurrent dropout 0.2 and activation function tanh. Next, the model utilizes convolutional layer 1D with 72 filters and 3x3 kernel size with the activation function ReLU. After convolutional layer, we have max pooling layer 1D of size 3x3 that is followed by a Simple RNN layer of size 64 with the activation function tanh, dropout 0.2 and recurrent dropout 0.2. The next layer is GRU layer of size 64 with recurrent dropout 0.2 and regularizer L1 and L2. The following layer is Dropout 0.2 and the last layer is a Dense layer of size depending on the number of labels to predict (for classes - 42, for superclasses - 9). The activation function used in the last layer is softmax to enable the multi-label classification.



Model 1- Simple RNN

Embedding (70)

Bidirectional Simple RNN (64, dropout = 0.1, recurrent_drpout = 0.20, activation = Tanh)

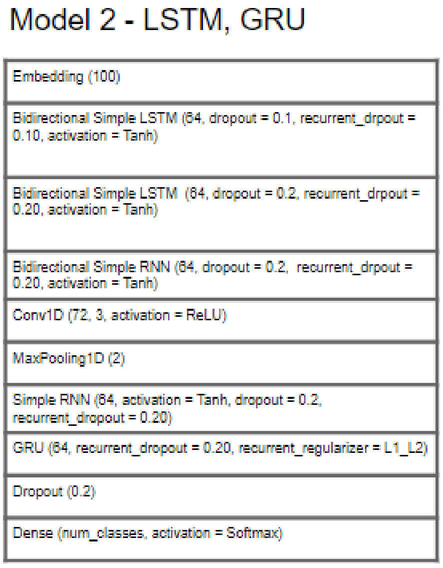
Bidirectional Simple RNN (64, dropout = 0.1, recurrent_drpout = 0.20, activation = Tanh)

Simple RNN (32, activation = Tanh)

Dropout (0.2)

Dense (num_classes, activation = Softmax)

Model 1 architecture.



Model 2 - LSTM, GRU

Embedding (100)

Bidirectional Simple LSTM (64, dropout = 0.1, recurrent_drpout = 0.10, activation = Tanh)

Bidirectional Simple LSTM (64, dropout = 0.2, recurrent_drpout = 0.20, activation = Tanh)

Bidirectional Simple RNN (64, dropout = 0.2, recurrent_drpout = 0.20, activation = Tanh)

Conv1D (72, 3, activation = ReLU)

MaxPooling1D (2)

Simple RNN (64, activation = Tanh, dropout = 0.2, recurrent_dropout = 0.20)

GRU (64, recurrent_dropout = 0.20, recurrent_regularizer = L1_L2)

Dropout (0.2)

Dense (num_classes, activation = Softmax)

Model 2 architecture.

4.3 Training Setting

The training data was splitted 80/20 and the testing batch stayed shuffled while the training data was ordered in the desired form explained in the next section. The text data was tokenized and padded for the set up of the embedding layers. The Keras Tokenizer was used to fit in the model. All the models have the same training setting. RMSProp is used as an optimizer and the loss function to compile the model is categorical cross-entropy function. The early stopping is used in all the experiments with batch size of 128 and 15 epochs. The training data is fed in the model at the same time. There is no setting of feeding the data gradually with more examples. This technique is widely used in Curriculum Learning but it was not utilized in our experiments. The training of the model was monitored by accuracy. The validation was not set up during the training since the model is fed with the ordered examples and the validation split would discard certain categories from the validation set and keep only few of them within the split. This would lead to the data imbalance and unreliable results.

4.4 Design Motivations

Using the same model setting and identical parameters, we tend to eliminate any variations within the training and model design. Thanks to this consistent setting we can investigate purely the impact of data ordering on the models performance. The experiments are solely focused on the idea of semantically clustered data examples. By "baseline" we refer to the baseline data ordering which is data randomly shuffled. This baseline is compared with the training examples specifically ordered for the sake of these experiments. The baseline data ordering was used to predict all classes, but was not ran for 9 superclasses manually created by the authors. One of the reasons to opt for the Simple RNN vs LSTM, GRU model comparison is to deal with text data. Although the predictions were done on rather short texts (headlines), the LSTM, GRU model2 is beneficial dealing with sequential data, such as text, i.a. sequences of characters. Using RMSProp as an optimizer is considered to be an effective extension of gradient descent and is preferred for fitting deep learning neural networks.

Future experiments planned to be completed involve using the data with short descriptions included in the training. For the experiments described here, we used only headlines to learn the model. Within almost 210k examples in the whole dataset, almost 20k examples do not have short descriptions of the articles. In order not to trim the training size, the predictions were done on the headlines since only 6 articles in the full dataset lack headlines.

5 Experimental Design

5.1 Semantic Clusters

The semantically related groups of news categories are called clusters. There are 42 distinct classes in the original dataset. These 42 categories were then manually clustered into 9 superclasses.

The superclass POLITICS includes the following classes: GENERAL POLITICS, THE WORLDPOST, WORLD NEWS, FIFTY, GOOD NEWS and U.S. NEWS.

The superclass HEALTH includes the following classes: WELLNESS, HEALTHY LIVING, GREEN and ENVIRONMENT.

The superclass LIFESTYLE includes the following classes: STYLE AND BEAUTY, FOOD AND DRINK, HOME AND LIVING, STYLE and TASTE.

The superclass ENTERTAINMENT includes

the following classes: LEISURE, TRAVEL, COMEDY, SPORTS and WEIRD NEWS.

The superclass FAMILY includes the following classes: PARENTING, PARENTS, WEDDINGS and DIVORCE.

The superclass CULTURE includes the following classes: MEDIA, ARTS, ARTS AND CULTURE and CULTURE AND ARTS.

The superclass COMMUNITIES includes the following classes: QUEER VOICES, BLACK VOICES, WOMEN and LATINO VOICES.

The superclass SOCIAL includes the following classes: CRIME, IMPACT, RELIGION, COLLEGE and EDUCATION.

The superclass SCIENCE TECH includes the following classes: BUSINESS, SCIENCE, TECH and MONEY.

The name of the POLITICS class in the original dataset was changed to GENERAL POLITICS to enable creation of the superclass POLITICS. In a similar way, the ENTERTAINMENT class from the original dataset was renamed as LEISURE to make ENTERTAINMENT a superclass category.

5.2 Experimental Data Orderings

The experimental design is strictly related to the way the training examples are ordered around the semantic clusters described in the section above. In order to demonstrate and validate the target contributions of data ordering to the NN training process, the data clusters were designed in the following way:

1. Alphabetic Clustering

   - ORDER 1: Ordering by superclass alphabetically. The training examples are sorted alphabetically starting from the superclass COMMUNITIES and going in the alphabetical order of other superclass label names.
   - ORDER 2: Ordering by class alphabetically. The training examples are sorted alphabetically starting from the class ARTS AND CULTURE and going in the alphabetical order of other class label names.

2. Semantic Clustering

   - ORDER 3: Clusters are grouped within one superclass in the descending order. It means that the first class in each superclass has the most number of texts in this class. Whereas, the last one class within the same superclass will have the least number of texts as training examples. All classes from one superclass are put together, i.a. their examples come next to each other during training but ordered based on the number of examples one class includes.

     This ordering mimicks the Curriculum Learning technique since it has the most examples for the model to learn from. It indicates that these instances are going to cause the least troubles in the evaluation phase. The model will have a chance to learn these examples better so it is easier to recognize and classify later. The classes that has the least number of examples are learnt later, just as the Curriculum Learning principle in deep learning states.
   - ORDER 4: Clusters are grouped within one superclass in the ascending order. It means that the first class in each superclass has the least number of texts in this class. Whereas, the last one class within the same superclass will have the most number of texts as training examples. All classes from one superclass are put together, i.a. their examples come next to each other during training but ordered based on the number of examples one class includes.

     This setting is the opposite of Curriculum Learning.
   - ORDER 5: Clusters are grouped within one superclass but they are not sorted according to the number of training examples within a class. The instances are randomly shuffled within one superclass.

6 Experimental Results

All the results of the experiments presented in this paper outperformed the baseline data ordering. This is a significant sign that data ordering had an impact on the model. All the predictions done on the superclasses made the model much more efficient in terms of performance. This finding is not that surprising when we compare it to the 42 classes that the model finds much more difficult to predict. Model 2 performed much better than model 1 which can be contributed to the size of the model. Although the scalability of the model 2 made the training time significantly longer, the performance boosted up. The best performing model within the classes predictions was ORDER4 that mimics Curriculum Learning. This means that the mentioned technique is very efficient in training the deep learning models and feeding the model with the easiest training examples first,

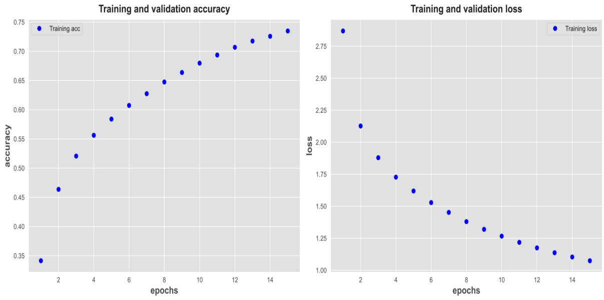


Figure 3: Training and Validation Accuracy and Loss of Model 2 on Order 1

followed by more difficult ones, brings beneficial results. Second best model was also clustered within one superclass but surprisingly, the data examples were shuffled within one cluster. This is a proof that semantically grouped training examples are better learnt by the model than randomly shuffled but not clustered in terms of their meaning. ORDER 1 had the poorest performance. This data ordering was not clustered semantically but only alphabetically to group together same news headlines within one superclass. Semantically clustered data did better than alphabetically ordered. Curriculum Learning seems to be the key component to data ordering techniques bringing best results to the models. As expected, LSTM and GRU layers improved the performance and helped to process long sequences and preserve the features of previous input at each time step. Model2 significantly outperformed the model1 in all the experiments. One surprising finding from the experiments is the performance of model1 on ORDER3. This is the only experiment when ORDER3 has the highest performance. However, it is still an ordering within semantic clusters but not alphabetical clusters which supports the hypothesis of semantically-related ordering.

	Model 1	Model 2
Baseline	49%	50%
Order 1	66.31%	73.49%
Order 2	66.77%	73.37%
Order 3	66.69%	73.56%
Order 4	**67.37%**	**74.04%**
Order 5	66.92%	73.46%

	Model 1	Model 2
Order 1	80.12%	84.12%
Order 2	80.62%	84.32%
Order 3	**80.93%**	85.01%
Order 4	80.16%	**85.26%**
Order 5	80.31%	84.98%

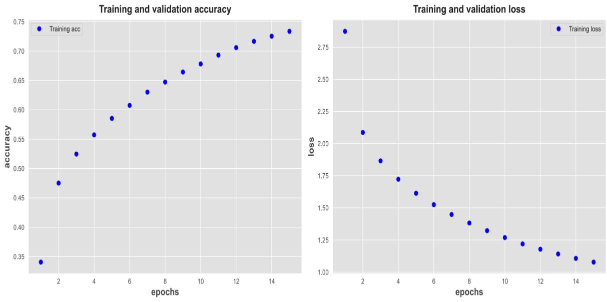


Figure 4: Training and Validation Accuracy and Loss of Model 2 on Order 2

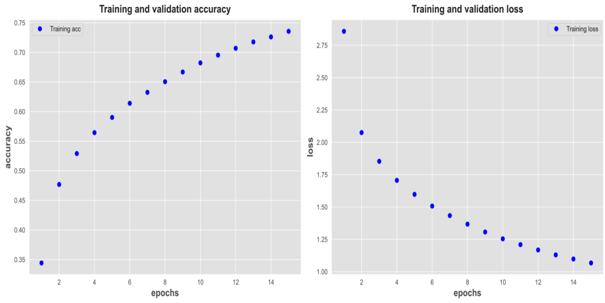


Figure 5: Training and Validation Accuracy and Loss of Model 2 on Order 3

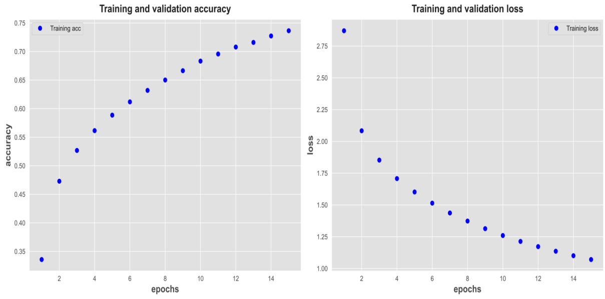


Figure 6: Training and Validation Accuracy and Loss of Model 2 on Order 4

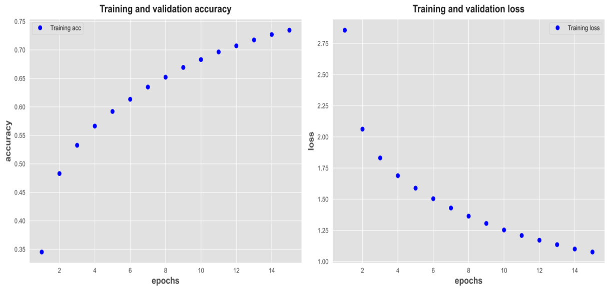


Figure 7: Training and Validation Accuracy and Loss of Model 2 on Order 5

7 Conclusions

Models trained on ordered datasets perform markedly better than their unordered counterparts. While it is evident that any intentional ordering improves model performance, and that peak performance for this experiment came from the semantically-ordered dataset, the difference in accuracy in ordered datasets is marginal. Future experiments should continue in evaluating what method of ordering a dataset is most effective.

We believe there are no ethical implications involved in the impact of this paper. The methods and findings regarding this experiment are fairly innocuous; the paper is based on re-ordering an open source dataset. We have no concerns of breaches of privacy, eschewing of accountability and transparency, or any negative social impact arising due to the findings this paper presents.

References

[1] Bender, E. M., and Koller, A. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 5185–5198.

[2] Cai, J., Utiyama, M., Sumita, E., and Zhang, Y. Dependency-based pre-ordering for Chinese-English machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Baltimore, Maryland, June 2014), Association for Computational Linguistics, pp. 155–160.

[3] Chang, E., Yeh, H.-S., and Demberg, V. Does the order of training samples matter? improving neural data-to-text generation with curriculum learning, 2021.

[4] Elman, J. Learning and development in neural networks: the importance of starting small. *Cognition 48* (08 1993), 71–99.

[5] Garg, S. Data ordering patterns for neural machine translation: An empirical study, 2019.

[6] Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision* (2020), Springer, pp. 466–483.

[7] Heo, H.-S., Jung, J.-w., Kang, J., Kwon, Y., Kim, Y. J., Lee, B.-J., and Chung, J. S. Self-supervised curriculum learning for speaker verification, 2022.

[8] Holmes, G., Frank, E., Fletcher, D., and Sterling, C. Efficiently correcting machine learning: considering the role of example ordering in human-in-the-loop training of image classification models. In *27th International Conference on Intelligent User Interfaces* (2022), pp. 584–593.

[9] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences 114*, 13 (2017), 3521–3526.

[10] Sezerer, E., and Tekir, S. Incorporating concreteness in multi-modal language models with curriculum learning. *Applied Sciences 11*, 17 (2021), 8241.

[11] Shao, C., and Feng, Y. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. *arXiv preprint arXiv:2203.03910* (2022).

[12] Shumailov, I., Shumaylov, Z., Kazhdan, D., Zhao, Y., Papernot, N., Erdogdu, M. A., and Anderson, R. J. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems 34* (2021), 18021–18032.

[13] Weinshall, D., Cohen, G., and Amir, D. Curriculum learning by transfer learning: Theory and experiments with deep networks, 2018.