When does deep multi-task learning work for loosely related document classification tasks?

Emma Kerinec
École Normale Supérieure
de Lyon
Lyon, France
emma.kerinec@ens-lyon.fr

Anders Søgaard
Dpt. of Computer Science
University of Copenhagen
soegaard@di.ku.dk

Chloé Braud*
Université de Lorraine,
CNRS, LORIA
Nancy, France
chloe.braud@loria.fr

Abstract

This work aims to contribute to our understanding of *when* multi-task learning through parameter sharing in deep neural networks leads to improvements over single-task learning. We focus on the setting of learning from *loosely related* tasks, for which no theoretical guarantees exist. We therefore approach the question empirically, studying which properties of datasets and single-task learning characteristics correlate with improvements from multi-task learning. We are the first to study this in a text classification setting and across more than 500 different task pairs.

1 Introduction

Multi-task learning is a set of techniques for exploiting synergies between related tasks, and in natural language processing (NLP), where there is an overwhelming number of related problems, and different ways to represent these problems, multi-task learning seems well-motivated. Since multi-task learning, by exploiting related tasks, also reduces the need for labeled data, multi-task learning is also often seen as a way to obtain more robust NLP for more domains and languages.

Multi-task learning has seen a revival in recent years, amplified by the success of deep learning techniques. Multi-task learning algorithms have been proven to lead to better performance for similar tasks, e.g., Baxter and others (2000), such as models of individual patients in health care, but recently multi-task learning has been applied to more loosely related sets of tasks in artificial intelligence. Examples include machine translation and syntactic parsing (Kaiser et al., 2017) or fixation prediction and sentence compression (Klerke, Goldberg, and Søgaard, 2016). Reported results have been promising, but in the case of loosely related tasks, often also with different label spaces, we have no guarantees that multi-task learning will work.

Recent studies have tried to study empirically *when* multi-task learning leads to improvements (Alonso and Plank, 2017; Bingel and Søgaard, 2017). These preliminary studies have argued – Bingel and Søgaard (2017) most clearly – that multi-task learning is particularly effective when the target task otherwise plateaus faster than the auxiliary task. This study compliments these studies, considering new tasks and architectures, and our findings are largely supportive of this conclusion. In text classification, however, performance also depends crucially on the divergence between the marginal distributions of words in the target and auxiliary task.

Document classification comes in many different flavors, including spam detection, sentiment analysis, customer support ticket routing, and diagnosis support based on patient records, but in this paper we focus on **topic-level multi-way classification**. We use the 20 Newsgroups dataset, a corpus of newsgroup posts that are labeled by the topics of the newsgroups. One key challenge in document classification is the high number of feature dimensions introduced by n-gram features, often outnumbering the number of document instances in the training corpus. Specifically, it is easy to overfit to the training corpus in high dimensions.

Multi-task learning (Caruana, 1993) has strong regularization effects and can therefore potentially make our models less prone to overfitting. Previous empirical meta-studies of multi-task learning have focused on sequence tagging problems and recurrent neural networks, but there is no guarantee that results extend to document classification. This work, which extends previous work on recur-

[*] This work was done, when the third author was affiliated with Dpt. of Computer Science, University of Copenhagen.

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 1–8
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

rent neural networks, is thus motivated by a) an interest in whether previous findings generalize to document classification algorithms – in our case, **multi-layered perceptrons**, b) a practical consideration that any recommendations coming out of a study of document classification would be helpful to a wider audience.

As already said, our focus on topic-level classification is motivated by the observation that this is an extremely common problem, and key to structuring content on websites, customer support ticket routing, intelligent email, etc. Also, the 20 Newsgroups corpus uses a set of 20 labels that are hierarchically organized (see Figure 1), which we can exploit to extract a large set of task pairs.

The problem that we consider is the following: *If we have two topic-level classification datasets that are loosely related – i.e, contrasts the same upper level classes in the hierarchy in Figure 1 – and we have run single-task experiments for each of these, when does multi-task learning help, keeping hyper-parameters fixed?* We approach this as a prediction problem, trying to predict gains or losses based on meta features such as dataset characteristics and features of the single-task learning curves. This approach was first introduced in (Bingel and Søgaard, 2017).

1.1 Contributions

Our contributions are as follows: a) We present the first study of when multi-task learning works in the context of document classification. b) This is, to the best of our knowledge, also the first meta-study that focuses on hard parameter sharing in multilayered perceptrons, although this approach to multi-task learning goes all the way back to (Caruana, 1993). c) We find that many of the results obtained with other types of deep neural networks scale to our case, but also that distributional divergence is strongly, negatively correlated with performance gains; something not observed with sequence tagging problems. Finally, we make all our code available at [anonymized].

2 Related Work

Document classification has a very long history and is one of the most fundamental applications of machine learning. It is extremely important to many industries, from customer support to medical diagnosis support.

The standard approach to document classification is to represent documents by what is known as *bags of words*, i.e., vector representations where each dimension encodes the presence or relative frequency of a particular n-gram (sequence of words). In this work, we use TF-IDF scores and only encode the presence of unigrams (words). Each document is thus a $|V|$-dimensional array of floats, where $|V|$ is the size of our vocabulary.

The dataset that we use, is 20 Newsgroups.[1] It has been used in several comparisons of classification algorithms (Dredze, Crammer, and Pereira, 2008; Crammer and Chechik, 2012), and some of the best results have been achieved with random forests and multi-layered perceptrons (deep learning models). The dataset, however, is also known to allow for over-fitting (Ribeiro, Singh, and Guestrin, 2016). Such overfitting can be remedied by multi-task learning. In this paper, we focus on multi-task learning with multi-layered perceptrons.

Multi-task learning comes in many different flavors, but most approaches can be cast as ways of doing matrix regularization. To see this, construct a $m \times n$ matrix for m models with n parameters. Multi-task learning corresponds to jointly fitting the m models penalized by a regularization term defined over this matrix. One common approach to multi-task learning, for example, is *mean-constrained* ℓ_2-regularization. The penalty in this case is the sum of the ℓ_2-distances of the m models to their mean.

In this paper, we focus on *hard parameter sharing*, in which we jointly learn m multi-layered perceptrons that share the parameters of their hidden layers. This is also the kind of architecture discussed in (Collobert et al., 2011), one of the seminal papers in multi-task learning for natural language processing. See Ruder (2017) for a more complete overview of multi-task learning algorithms used in natural language processing.

Hard parameter sharing comes with several guarantees when applied to closely related tasks (Baxter and others, 2000), including a reduction in Rademacher complexity (Maurer, 2006). These guarantees, however, do not apply to our case of more loosely related tasks. For example, (Baxter and others, 2000) requires the tasks to have shared optimal hypothesis classes; which does not have to be the case in 20 Newsgroups.

[1] http://qwone.com/~jason/20Newsgroups/

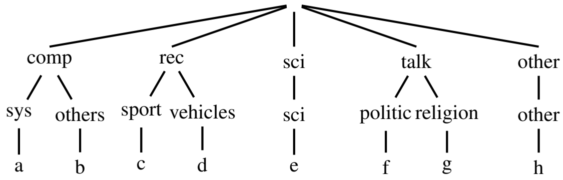


Figure 1: Hierarchical structure of 20 Newsgroups, with a= ibm.pc.hardware, mac.hardware; b= graphics, os.ms-windows.misc, windows.x; c= baseball, hockey; d= autos, motorcycles; e= crypt, electronics, med, space; f= misc, guns, mideast; g= misc, atheism, christian, h= forsale.

3 Methodology

We begin with a brief summary of our methodology: We sample pairs of tasks from 20 Newsgroups. The documents are represented as TF-IDF vectors, and we train single-task and multi-task multilayered perceptrons to predict topics from such vectors. We then run meta-experiments using logistic regression classifiers to predict the sign of the relative difference between multi-task and single-task performance, from features derived from the data and the single-task runs. We are primarily interested in the coefficients of the logistic regression meta-models, which tell us what characteristics of the data and the single-task experiments are predictive of multi-task learning gains.

3.1 20 Newsgroups

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned across 20 different topics. It contains about 60,000 different words in total.

Some of the newsgroups are very closely related and can be seen as subtopics of the same topic, while others are highly unrelated. The topics can be represented as a 3-level hierarchy: The first level partitions the set of topics into 5 classes (e.g. `comp, rec...`), the second one into 8 subclasses (e.g. `sys, others, sport...`), and at the leaf nodes we have the 20 topics (e.g. `ibm.pc.hardware, baseball...`); see Figure 1.

3.2 Classification tasks

Based on the 20 Newsgroups' structure, we define pairs of tasks in ways similar to previous studies (Søgaard and Johannsen, 2012). We do this in two different ways, leading to Problem 1 and 2, defined below.

3.2.1 Problem 1 (RELATED TOPICS)

The main task is to distinguish between two topics `A` and `B` (third level) that have the same ancestor at the first level of the above hierarchy, i.e. they pertain to the same class, but to different subclasses. An auxiliary task is to distinguish between two topics `C` and `D`, with the following constraints: `C` has the same father as `A`, and `D` the same as `B`.

A task pair example would be: A=`baseball` and B=`autos` for the main task since; C=`hockey` and D=`motorcycles` for the auxiliary task (see 2). We obtained 52 unique such pairs of main-auxiliary tasks for problem 1.

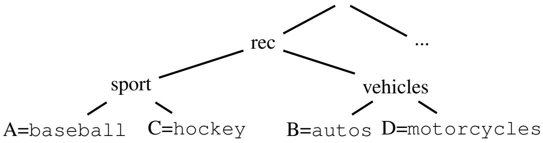


Figure 2: Problem 1 (Related topics): A and B are the main tasks, C and D the auxiliary ones.

3.2.2 Problem 2 (UNRELATED TOPICS)

For the second problem, we keep the constraints that `C` has the same father as `A` and `D` the same as `B`, and that `A` and `B` have different fathers. However, `A` and `B` are not forced to have the same ancestor at the first level anymore. In this setting, the main and auxiliary tasks could be about distinguishing texts corresponding to unrelated topics, but they still share topics pertaining to the same classes, making multi-task learning a relevant framework.

An example of pairs of tasks would be: A=`guns` and B=`autos` for the main task; C=`Mideast` and D=`motorcycles` for the auxiliary task (see Figure 3).

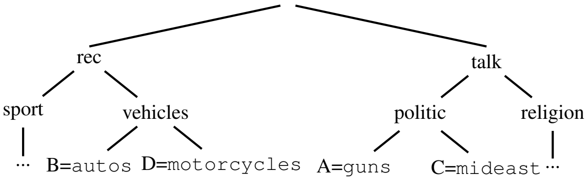


Figure 3: Problem 2 (Unrelated topics):A and B are the main tasks, C and D the auxiliary ones.

We obtained 516 different pairs of main-auxiliary tasks for UNRELATED TOPICS.

Note that the instances (i.e. pairs of main-auxiliary tasks) of RELATED TOPICS are included in the set of instances of UNRELATED TOPICS. We have many more instances for UNRELATED

TOPICS than for RELATED TOPICS, which means that we have many more training points when trying to predict the performance of multi-task learning.

3.3 Representation of the data

We use TF-IDF (term frequency-inverse document frequency) over the bag-of-words to represent the data. The TF-IDF value increases proportionally to the number of times a word appears in a document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. This representation is known to be efficient (Salton and Buckley, 1988; Aizawa, 2003); especially in the case of text classification (Zhang, Yoshida, and Tang, 2011). We keep the 10,000 most frequent features, the frequency being computed on the training data available for the entire 20 Newsgroups corpus.

3.4 Models

Both our single and multi-task learning architectures consist of a multi-layered perceptron with two hidden layers. In the case of multi-task learning, those layers are shared across all tasks. This setting is known as hard parameter sharing. Hard parameter sharing was first introduced by (Caruana, 1993) and used with success for different tasks, for example in (Collobert et al., 2011; Klerke, Goldberg, and Søgaard, 2016; Plank, Søgaard, and Goldberg, 2016). Hard parameter sharing greatly reduces the risk of overfitting. In fact, Baxter and others (2000) showed that the risk of overfitting the shared parameters is an order n where n is the number of tasks smaller than overfitting the task-specific parameters, i.e. the output layers.

The input is thus a 10,000-dimensional TF-IDF vector representation of the texts. A training step consists of sampling a random batch of 32 instances, i.e. texts (for both main and auxiliary task in the case of multi-task learning) and minimizing the binary cross-entropy loss using an Adam optimizer (Kingma and Ba, 2014).

We tune the following hyper parameters of the single-task architectures on a similar document classification problem, using data from Amazon reviews,[2] and, following (Bingel and Søgaard, 2017), we apply *the same* hyper-parameter values to multi-task learning: number of hidden layers (2) and layer size (100). See §4.1 for number of epochs (100).

3.5 Meta-analysis

We want to investigate whether we can predict gains from multi-task learning given features of the data sets and single-task learning characteristics, as well as understand how gains correlate with data set and single-task learning characteristics. For each problem instance, we thus extract several features from the datasets and the learning curves of the single task models. These features are similar to those used in (Bingel and Søgaard, 2017):

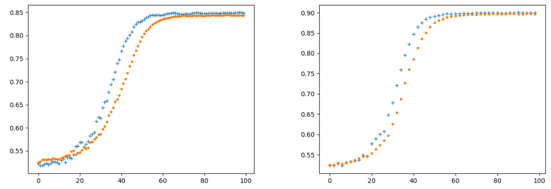
- Jensen-Shannon Divergence between the (unigram) word distributions of the target and auxiliary task training sets, as well as internally (between target and test data) for each task,
- Gradients of the loss curve at 10, 25, 50 and 75 percent of a training of 150 epochs, for each single-task, as well as the relative differences in the learning curve gradients,
- Type-token ratios and out-of-vocabulary rates in the target and auxiliary task training sets, and their relative difference,
- Finally, we fit logarithmic functions to the (log-like) loss curves, where the function is of the form: $a \cdot ln(c \cdot i + d) + b$, and we include a and c as features. Both parameters relate to the steepness of the loss curve, reflecting when training plateaus or comes with diminishing returns.

In total, for each problem instance we have 30 features that we normalize to the $[0, 1]$ interval. We use logistic regression to predict benefits or detriments of multi-task learning setups based on the features computed above.

4 Experiments

We run single-task and multi-task learning experiments for all pairs of main and auxiliary tasks, as described in Section 3.2. We then extract data characteristics and features from the logs of the single-task learning experiments. We train a meta-learning model to predict gains from doing multi-task learning over single-task performance using

[2] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html



(a) RELATED TOPICS (b) UNRELATED TOPICS

Figure 4: Mean F_1 over the number of epochs, for single-task (crosses/blue) and multi-task learning models (points/orange), for classification problems 1 and 2.

the above features. Then we build a final model to predict gains from multi-task learning using these pairs as instances. We use the 20 Newsgroups for both RELATED TOPICS and UNRELATED TOPICS, as explained above. We use 200 topics for each class for training, and the rest of each dataset for testing (5-700 data points, depending on the topics).

4.1 Hyper-parameters

Hyper-parameters were tuned using the Amazon data, as described in §3.4. Our models are trained with two layers of size 100. The input is a 10,000 dimensional TF-IDF vector, and the output is a probability distribution from a softmax layer, whose predictions are evaluated using cross-entropy loss.

Figures 4a and 4b plot the impact of the number of epochs on the F_1 scores. This parameter was not optimized on the Amazon data, but set such that multi-task learning gains were reasonably balanced.

In meta-learning, when predicting the gains from multi-task learning, we use the mean performance of 100 runs of randomized five-fold cross-validation with logistic regression.

4.2 Evaluation

We train single-task models for all tasks, as well as multi-task learning models for all combinations of target and auxiliary tasks. We report the F_1 gains obtained for multi-task learning over single-task learning below.

Our real aim, however, is to try to predict the gains one can get from doing multi-task learning. This is a meta-learning problem, and here, the above experiments are our instances, i.e., one instance for each of the main-auxiliary task pairs, meaning that we have 52 instances for RELATED TOPICS and 516 for UNRELATED TOPICS. In order to compensate for the small number of training instances, we repeat our RELATED TOPICS experiments five times with random initializations, and report means over the results. We use the same procedure for UNRELATED TOPICS, also. F_1 scores, obtained by a logistic regression model over 100 runs using a 5-fold cross-validation procedure, are reported at the end of the next section.

5 Results

We first discuss the performance of our multi-task learning models on the 20 NEWSGROUPS data, and then present the results of our meta-learning experiments.

5.1 Multi-task versus single-task learning

As mentioned above, we report averages over five runs. The mean F_1 scores across all the problems, and five runs, are presented in Table 1. We observe that on average, multi-task learning leads to slight improvements over single-task learning. This holds for both our problems, also for RELATED TOPICS. The number of epochs needed to train the multi-task models is slightly greater than the one for the single-task ones (Figures 4a and 4b), and the global stabilization occurs after approximatively 75 epochs. We can also observed that UNRELATED TOPICS, where tasks to differentiate are in general theoretically more different, has better result than RELATED TOPICS (for both single-task and multi-task learning) see Table 1.

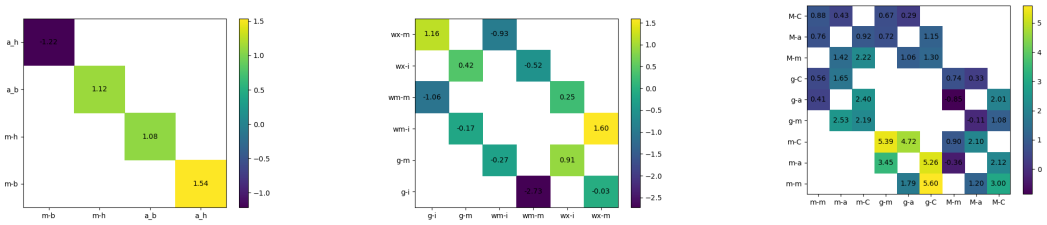
For RELATED TOPICS, we see improvements in more than 70% of the cases, and the mean gain is about 5%. Figure 5a presents the relative gains and losses over the different high-level classes of the RELATED TOPICS problem. Note there is a lot of variance. Some class pairs exhibit a lot of synergy, with gains doing multi-task learning, while others seem relatively immune to multi-task learning. For UNRELATED TOPICS, multi-task learning leads to improvements in about 57% of all cases.

5.2 Predicting gains from multi-task learning

In our meta-learning experiment, the objective is to predict multi-task learning gains given the dataset and single-task learning characteristics. This is not only because it is of practical importance to be able to predict whether multi-task learning is worthwhile, when dealing with massive

	Single-task	Multi-task	Improvements
Related	0.834	0.843	0.719
Unrelated	0.893	0.897	0.572

Table 1: Mean F_1 score for single-task and multi-task models, with average fraction of datasets with improvements.



(a) ”rec”: Vehicles (motorcycles and autos) vs. Sports (hockey and baseball).

(b) ”comp”: Others (graphics, miscellaneous, Windows) vs. Systems (IBM, Mac).

(c) ”talk”: Politics (miscellaneous, guns, Middle East) vs. Religion (miscellaneous, atheism, Christianity).

Figure 5: Relative F_1 gains from multi-task learning for *Related Topics*

datasets or thousands of tasks. More importantly, our meta-learning models implicitly learn correlations between such characteristics and gains, giving us insights as to *when* and *why* multi-task learning works. If a dataset characteristic, for example, is highly predictive of gains, this can either be a feature that puts single-task learning at a disadvantage, or something that multi-task learning can exploit.

The mean scores over 100 runs (5-fold CV) of our logistic regression model for different feature combinations are listed in Table 2. The results show that generally, features extracted from the loss curves are more predictive of gains than any other features. This confirms findings in Bingel and Søgaard (2017).

	Related Topics	Unrelated Topics
Using all features	0.67	0.57
Not using curve features	0.66	0.53
Only using curve features	0.71	0.58
Only using ratio features	0.69	0.57

Table 2: Mean performance across 100 runs of 5-fold CV logistic regression.

6 Discussion

The mean score (inverse rank) of each predictor is given in Table 4a; and the coefficients of the predictors in Table 4b. The JSD features either capture divergences between target and auxiliary tasks, in general, or between the classes, or between target and auxiliary with respect to either positive or negative class. Other features include the number of words in the training and test set, their relative numbers, or the relative numbers between target and auxiliary tasks (equivalent to type-token ratios). Finally, the curve-related features come in two flavors. One set is simply the gradients of the loss curve at different time steps. The other set is the parameters a and c from a log-curve fitted to the entire loss curve.

6.1 Most predictive features

The most predictive features across both tasks are Jensen-Shannon divergences, and the fitted loss curve parameters a and c. OOV rate is also predictive of gains, i.e., correlated with gains from multi-task learning, which makes sense, since our embedding parameters are updated during training, leading to better representations for rare words that occur more frequently in the auxiliary data.

Jensen-Shannon Divergence (JSD) We compute JSD between training and test, in both tasks, and their relative ratio, as well as between classes. JSD between training and test is strongly negatively correlated with gains from multi-task learning. In other words, the more divergence between your target and your auxiliary task, the *less* likely multi-task learning is to work. The importance of JSD is very interesting – and per-

Feature	Data	Inverse rank
JSD pos. class	main	23
Curve param a	main	21
JSD pos. class	ratio	21
Curve gradient 10%	main	20
Curve gradient 10%	ratio	18
JSD between classes	aux	17
# words	ratio	17
OOV rate	all	17
Curve param c	aux	16
Curve gradient 50%	ratio	16
JSD neg. class	aux	16
# words	main	15
Curve gradient 75%	main	14
Curve gradient 25%	aux	14
JSD between classes	ratio	14
Curve gradient 75%	ratio	14
JSD neg. class	all	14
Curve gradient 25%	ratio	13
Curve gradient 50%	aux	12
Curve gradient 75%	aux	12
Curve param a	aux	11
Curve param a	ratio	11
# words	test	11
Curve gradient 50%	main	10
Curve param c	ratio	10
JSD pos. class	all	10
Curve param c	main	9
# words	aux	9
Curve gradient 10%	aux	9
Curve gradient 25%	main	8

(a) Inverse ranks for RELATED TOPICS

Feature	Data	Coefficient
JSD pos. class	all	-0.93
JSD neg. class	all	-0.88
OOV rate	all	0.81
JSD between classes	all	0.64
JSD between classes	aux	0.63
JSD between classes	main	0.58
# words	test	-0.49
# words	train	-0.47
Curve param a	ratio	0.34
Curve param a	aux	-0.31
Curve gradient 75%	ratio	0.26
Curve param c	ratio	0.24
# words	aux	-0.21
Curve param c	main	-0.17
Curve gradient 75%	main	0.17
# words	main	0.13
Curve gradient 50%	aux	-0.11
Curve gradient 75%	aux	0.10
JSD neg. class	aux	-0.08
Curve gradient 50%	main	-0.07
Curve param a	main	0.07
JSD pos. class	aux	0.07
Curve gradient 25%	aux	-0.05
Curve gradient 10%	ratio	0.04
Curve gradient 25%	ratio	0.04
Curve gradient 25%	main	0.03
Curve gradient 50%	ratio	-0.03
Curve gradient 10%	aux	-0.02
Curve param c	aux	-0.02
Curve gradient 10%	main	0.01

(b) Coefficients for UNRELATED TOPICS

Table 3: Average inverse ranks and average logistic regression coefficients of various predictors of gains from multi-task learning

haps a bit surprising in the light of recent results for sequence tagging (Alonso and Plank, 2017; Bingel and Søgaard, 2017). These recent results suggested that JSD is not predictive of multi-task learning performance *at all*. Of course, JSD over unigram occurrences is more closely related to the model bias arising when training document classification models on loosely related tasks, than to the model bias in sequence models. After all, transition probabilities are typically at least as important as emission probabilities in statistical sequence tagging models.

Loss curve gradients were shown in (Bingel and Søgaard, 2017) to be the best predictors of multi-task learning gains. The intuition offered there is that multi-task learning is more likely to work when the target task quickly plateaus, but the auxiliary task keeps pounding, eventually letting the target task out of a potentially suboptimal local optimum. Multi-task learning leads to a smoother loss landscape, where it is harder to get trapped, and when randomly sampling from the auxiliary task, also, there is ample chance to be led out of poor, local optima. Note that in our experiments the good predictors based on loss curve gradients are found in the last regions of the curve, just before early stopping.

Stability Some features are highly correlated, which can produce instability – and poor results and misleading coefficients – when training logistic regression models. Note, however, that we report averages over multiple models. This is similar to the idea of using *stability selection* (Meinshausen and Bühlmann, 2010), though averaging over multiple problems is arguably more robust than doing it over bootstrap samples with replacement.

7 Conclusion

We have investigated the performance of single-task and multi-task multi layer perceptrons for text classification using a TF-IDF representation of documents. We ran experiments on the 20 Newsgroups corpus and took advantage of the class hierarchy in this dataset, to extract hundreds of pairs of loosely related documents, for which no theoretical guarantees exist.

Based on this data, we conduct meta-learning experiments, trying to predict when multi-task learning works, and when it does not. We inspect the coefficients of such meta models to estimate the contribution of various dataset features or learning characteristics to such gains. Our experiments show the importance of loss curve gradients and out-of-vocabulary rates, supporting recent findings from sequence tagging (Bingel and Søgaard, 2017), but we also see that biases in the marginal distribution of the data, as measured by JSD, are predictive of multi-task learning gains in document classification.

References

Aizawa, A. 2003. An information-theoretic perspective of tfidf measures. *Information Processing and Management*.

Alonso, H. M., and Plank, B. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL*.

Baxter, J., et al. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research (JAIR)* 12:3.

Bingel, J., and Søgaard, A. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *EACL*.

Caruana, R. 1993. Multitask learning: a knowledge-based source of inductive bias. In *ICML*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Crammer, K., and Chechik, G. 2012. Adaptive regularization of weight matrices. In *ICML*.

Dredze, M.; Crammer, K.; and Pereira, F. 2008. Confidence-weighted linear classification. In *ICML*.

Kaiser, L.; Gomez, A.; Shazeer, N.; Vaswani, A.; Parmar, N.; Jones, L.; and Uszkoreit, J. 2017. One model to learn them all. In *https://arxiv.org/abs/1706.05137*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klerke, S.; Goldberg, Y.; and Søgaard, A. 2016. Improving sentence compression by learning to predict gaze. In *NAACL*.

Maurer, A. 2006. Bounds for linear multi-task learning. *Journal of Machine Learning Research* 6:117–139.

Meinshausen, N., and Bühlmann, P. 2010. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72(4):417–473.

Plank, B.; Søgaard, A.; and Goldberg, Y. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *ACL*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should I trust you - explaining the predictions of any classifier. In *NAACL*.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *CoRR*.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*.

Søgaard, A., and Johannsen, A. 2012. Robust learning in random subspaces: equipping NLP for OOV effects. In *COLING*.

Zhang, W.; Yoshida, T.; and Tang, X. 2011. A comparative study of tf*idf, lsi and multi-words for text classification. *Expert Systems with Applications*.