Recurrent Neural Network Language Model Adaptation with Curriculum Learning

Yangyang Shi, Martha Larson, Catholijn M. Jonker
Department of Intelligent Systems, Delft University of Technology
Mekelweg 4, 2628CD Delft, The Netherlands. yangyangshi@ieee.org

Abstract

This paper addresses the issue of language model adaptation for Recurrent Neural Network Language Models (RNNLMs), which have recently emerged as a state-of-the-art method for language modeling in the area of speech recognition. Curriculum learning is an established machine learning approach that achieves better models by applying a curriculum, i.e., a well-planned ordering of the training data, during the learning process. Our contribution is to demonstrate the importance of curriculum learning methods for adapting RNNLMs and to provide key insights on how it should be applied. RNNLMs model language in a continuous space and can theoretically exploit word-dependency information over arbitrarily long distances. These characteristics give RNNLMs the ability to learn patterns robustly with relatively little training data, implying that they are well suited for adaptation. In this paper, we focus on two related challenges facing language models: *within-domain adaptation* and *limited-data within-domain adaptation*. We propose three types of curricula that start with general data, i.e., characterizing the domain as a whole, and move towards specific data, i.e., characterizing the sub-domain targeted for adaptation. Effectively, these curricula result in a model that can be considered to represent an implicit interpolation between general data and sub-domain-specific

data. We carry out an extensive set of experiments that investigates how adapting RNNLMs using curriculum learning can improve their performance.

Our first set of experiments addresses the within-domain adaptation challenge, i.e., creating models that are adapted to specific sub-domains that are part of a larger, heterogeneous domain of speech data. Under this challenge, all training data is available to the system at the time when the language model is trained. First, we demonstrate that curriculum learning can be used to create effective sub-domain-adapted RNNLMs. Second, we show that a combination of sub-domain-adapted RNNLMs can be used if the sub-domain of the target data is unknown at test time. Third, we explore the potential of applying combinations of sub-domain-adapted RNNLMs to data for which sub-domain information is unknown at training time and must be inferred.

Our second set of experiments addresses limited-data within-domain adaptation, i.e., adapting an existing model trained on a large set of data using a smaller amount of data from the target sub-domain. Under this challenge, data from the target sub-domain is not available at the time when the language model is trained, but rather becomes available little by little over time. We demonstrate that the implicit interpolation carried out by applying curriculum learning methods to RNNLMs outperforms conventional interpolation and has the potential to make more of less adaptation data.

Keywords: Recurrent Neural Networks; Language Models; curriculum Learning; Latent Dirichlet Allocation; Topics; Socio-situational setting.

---

1. Introduction

The task of statistical language models is to judge whether a sequence of words is well-formed or not. Conventional n-gram language models factorize the joint probabilities of all the words in a sequence into a product of probabilities of each word given information about its history, i.e., the preceding $n-1$ words. By using word histories, n-gram language models capture local regularities of languages. However, n-gram language models can only exploit an n-gram if the exact string of n words is present in the training data. As n grows large, the chance that an n-gram seen in the target data was also present in the training data falls off sharply. For this reason, conventional n-gram language models easily suffer from data sparseness. In practice, the history length $n-1$ that can be effectively exploited is quite limited. For this reason, n-gram language models lack adequate means to model long-distance dependencies.

These known shortcomings are addressed by Recurrent Neural Network Language Models (RNNLMs). Recently, RNNLMs have been demonstrated to outperform n-gram language models for speech recognition (Mikolov et al., 2010). Their superior capabilities rely on two mechanisms. First, RNNLMs map the discrete word-based vocabulary into a continuous space. As a result, the model can exploit word sequences which are similar, without requiring them to be exactly identical. This mechanism helps to reduce the effect of data sparseness. Second, RNNLMs are explicitly equipped to handle long-distance dependencies. The recurrent loop in their architecture constitutes a memory that allows them to model arbitrarily long word histories theoretically.

In this paper, we investigate language model adaptation for RNNLMs, and specifically address two central challenges for language model adaptation, *within-*

domain adaptation and *limited-data within-domain adaptation*, originally identified by Rosenfeld (1994) and explained later in depth. The main contribution of this paper is to demonstrate that curriculum learning is an important technique for carrying out the adaptation of RNNLMs and to provide insights on how it must be applied in order to be effective for improving speech recognition.

Curriculum learning applies a specific, well-planned ordering of the training data, referred to as a 'curriculum', during the learning process and is an established approach in the machine learning community. When conventional n-gram language models are trained, the order in which the training data is processed has no impact on the outcome of the training process. In contrast, Neural Networks are indeed sensitive to the differences in the order in which the training data is presented to them. The work of Bengio et al. (2009) attributes the benefits of curriculum learning in Neural Network training to an ability of the curriculum to guide the learner, in particular, directing it away from inappropriate local minima and towards more suitable ones.

The advantages that curriculum learning offers to RNNLMs for speech recognition have been previously established in the literature (Mikolov et al., 2010, 2011b). The previous work has focused on dynamically updating language models during the recognition process (Mikolov et al., 2010) and in optimal reduction and sorting of the training data (Mikolov et al., 2011b). The existing work points out that training data presented later in the training process has more influence on the final form of the model than the initial part of the training data. As such, curriculum learning can be used to accomplish an implicit interpolation of the training data, where certain parts of the data is given more importance than others.

The specific issue of adaptation is particularly important for RNNLMs, and as such deserves dedicated attention. A key reason for its importance is the relatively high cost of training RNNLMs, which can be attributed to a range of factors. Here, we mention in particular the fact that the whole training set is usually presented to the model multiple times (referred to as 'training epochs'). The relatively high cost of the training phase of RNNLMs means that retraining the language model whenever new training data becomes available is prohibitively costly.

Curriculum learning has several distinction advantages to offer for the adaptation of RNNLM to specific sub-domains. First, curriculum learning provides a method to effectively carry out implicit interpolation that does not require the parameter optimization needed for conventional interpolation. Second, as has been pointed out by Iyer et al. (1994), Iyer and Ostendorf (1999), and, more recently, by Mikolov and Zweig (2012), one danger of adaptation models that build individual component models on data sub-sets is fragmentation. Fragmentation refers to the fact that as more and more component models are built, relatively less data is available to train each. Using curriculum learning to train sub-domain adapted RNNLMs neatly circumvents the fragmentation issue. All training data can be used to train each adapted model; it is the order in which the data is presented to the model during training that makes the difference. In short, although the potential of curriculum learning for RNNLMs has been established and offers clear advantages, the challenges of curriculum learning to language model adaptation for RNNLMs remains nearly entirely unaddressed. The purpose of this paper is to fill that gap.

The key insight of this paper is that in order to carry out adaptation, curriculum learning should be used to train language models from general patterns, characteristic of the training data as a whole, to specific patterns characteristic of one

specific sub-domain of the overall data. The move from general patterns to specific patterns can be viewed as a special case of a move from simple patterns to complex patterns, discussed in the literature, e.g., by Elman (1993).

We propose three types of curricula created with three different strategies: Start-from-Vocabulary (*Start-Vocab*), Data Sorting (*Data-Sort*) and All-then-Specific (*All-Specific*). Our experiments are designed to give insight into which types of curriculum learning are best in which situations and which other factors influence the performance of curriculum learning for RNNLM adaptation. When RNNLMs are applied in practice to improve speech recognition, they are usually applied to the task of rescoring the N-best output produced by a speech recognizer that has carried out an initial pass over the spoken data. For this reason, all our experiments either produce word prediction results or rescoring results.

Our first set of experiments addresses the challenge of *within-domain adaptation*. As characterized by Rosenfeld (1994), within-domain adaptation can be used to deal with heterogeneous data sets that contain different sub-domains. Different sub-domains are characterized by different word-usage patterns, which are, in turn, reflected by different n-gram distributions. For example, in the Spoken Dutch Corpus which is described in detail later, different speech styles (e.g., spontaneous conversation, lecture and read speech) form different sub-domains. Note that we focus our investigation on within-domain adaptation rather than cross-domain adaptation, since, as mentioned by Rosenfeld (1994), it is relatively rare that a speech recognizer would be trained on one domain and used in a different one. Under the within-domain adaptation challenge, the totality of the training data becomes available at the same time, and the goal is to create models that are adapted to specific sub-domains.

Our first within-domain adaptation experiment investigates the oracle situation in which sub-domain information is known during both training and testing. This experiment demonstrates that sub-domain-adapted RNNLMs indeed outperform both general RNNLMs as well as RNNLMs that have been adapted to the sub-domain using linear interpolation between the general model and a sub-domain specific model. The second experiment investigates the situation in which sub-domain information is known during training, but unknown during testing. Here, we investigate two methods of combining sub-domain-adapted RNNLMs. The third experiment investigates the situation in which no sub-domain information is known for either training or testing. We use Latent Dirichlet Allocation with k-means clustering (Qiu and Xu, 2013) on the sentence level to automatically build sub-domains in the training data. Curriculum learning is applied to create sub-domain-adapted RNNLMs for each sub-domain, which are combined using the combination methods in the previous experiment.

Our second set of experiments addresses the challenge of *limited-data within-domain adaptation*. This challenge corresponds to the situation often faced by speech recognizers: a relatively large amount of data is available to train an initial model, and, with time, more and more data becomes available that can be used to update the model. The new data is from the same domain, but can be considered to be from a different sub-domain because of shifts in data characteristics over time. Here, we test a sub-domain adapted RNNLM, but focus particularly on the fact that the amount of adaptation data is limited and also on the fact that the vocabulary of the adaptation data is unknown at the time of training of the initial model.

The rest of the paper is organized as follows. Section 2 discusses related work in the areas of adaptive language modeling, curriculum learning and other ad-

vanced language models such as RNNLMs, mixture models and class based language models. In Section 3, we present the three types of curriculum learning that we use to train the sub-domain-adapted RNNLMs. We then present the experiments addressing the challenge of *within-domain adaptation* (Section 4) and the challenge of *limited-data within-domain adaptation* (Section 5). The final section concludes and presents an outlook.

2. Related Work

The approaches presented in this paper related to previous work that has been carried out in areas of adaptive language modeling, curriculum learning for recurrent neural networks, and recurrent neural network language modeling, are covered in this section in turn.

2.1. Adaptive Language Modeling

Approaches to the adaptation of statistical language models have been characterized by Bellegarda (2004) as belonging to three categories: model interpolation, constraint specification, and topic information. We discuss each category and mention its relationship to the approach proposed in this paper.

Cache-based statistical language models (Kuhn and de Mori, 1990; Jelinek et al., 1991) carry out dynamic adaptation during the process of recognition and fall under the category of model interpolation. These models represent one of the initial attempts to model long-distance dependency in language models. Basically, they involve the linear interpolation of an n-gram model and a dynamic cache component model. These models are based on the assumption that a word used recently has bigger probability than its overall probability. Cache-based statistical language models face the challenge of uncertainty of the adaptation data

that is introduced by speech recognition errors occurring in the window of recent words that is used as the cache. This challenge was mentioned by Mikolov et al. (2010) in their investigation of dynamic RNNLMs, which continue to update their weights during the test process. More recently, Mikolov and Zweig (2012) proposed a dynamically updating RNNLM that uses an LDA representation of recently recognized words in order to increase the amount of topic-related context information exploited by the language model. We limit our treatment of cache-based language models to this relatively brief mention, since our focus here is set on language model adaptation methods that are applied during training rather than during testing.

Constraint specification methods are mainly associated with maximum entropy language models. The process by which the constraint features are integrated into the model during training is forced to respect the maximum entropy criterion. Rosenfeld (1996) showed that maximum entropy language models, using trigger and n-gram features, achieve a significant improvement over n-gram language models in terms of perplexity and word error rate. Since their introduction, maximum entropy language models have enjoyed substantial success in adaptive statistical language modeling (Berger et al., 1996; Rosenfeld et al., 2001; Amaya and Benedí, 2001; Chen, 2009). Actually, from the neural network language modeling perspective, the maximum entropy model can be viewed as a neural network language model with no hidden layer. In this paper, some of the RNNLMs are built with maximum entropy extensions (Mikolov et al., 2011b).

Topic information methods for language model adaptation include topic-based language models and mixture models. Many of these models employ a model interpolation strategy, but are discussed here as topic information models rather than

interpolation models because they attempt to incorporate explicit representation of topics.

The topic-based language models of Gildea and Hofmann (1999) can be viewed as variants of class-based language models (Brown et al., 1992). These models map the words in the vocabulary to a smaller number of classes. Topic-based language models use the same mathematical formulation as class-based models, except that in topic-based models the topic is treated as a hidden variable that is calculated according to the expectation maximization algorithm. Exploiting the same basic insight, topics have been treated as hidden nodes in Dynamic Bayesian Networks (Wiggers and Rothkrantz, 2006; Shi et al., 2010). In Tam and Schultz (2005), Latent Dirichlet Allocation (LDA) language models were proposed. These models integrate topic information by interpolating the n-gram model with LDA unigram. Specifically, the LDA unigram is a combination of the topic probabilities with a conditional probability of present word given the latent topics which were estimated by LDA (Blei et al., 2003).

In the work of Iyer et al. (1994) and Iyer and Ostendorf (1999) on topic mixture models, the joint probability of each sentence is calculated as a linear interpolation of the sentence probabilities from k component language models, P_k, each modeling a separate topic,

$$P(s) = \sum_k \lambda_k P_k(s) = \sum_k \lambda_k \prod_i P_k(w_i|h(w_i)). \quad (1)$$

Here, $h(w_i)$ is the history of w_i in sentence s. The component models are trained on sub-sets of the training data, each containing the data that corresponds to an individual topic. The sub-sets are obtained by clustering the training data. In (Iyer and Ostendorf, 1999), a two-stage clustering process is used to partition the

data. The quality of the clustering, i.e., the suitability of the resulting partitions, is essential for the performance of the model.

Our approach exploits many of the same mechanisms as this work. We use a mixture of component models, as in Eq. (1), as a *soft decision method* to combine individual sub-domain adapted RNNLMs. Because the sub-domain-adapted RNNLM is able to make a highly accurate prediction of the sub-domain, we compare the topic mixture model with a *hard decision method*. Instead of using interpolation, this method makes a definitive decision about the sub-domain to which a given sentence belongs, and uses the corresponding sub-domain-adapted RNNLM to score that sentence.

We also use a clustering method in order to create sub-domains in cases in which sub-domain information is not available in the training data. However, instead of using a two-stage clustering method, we make use of LDA. Our choice is motivated by the status of LDA as the state of the art in topic representation (Blei et al., 2003). We form sub-domains by applying k-means clustering to LDA-based topic representations of the sentences in our data.

Finally, we would like to point out that our sub-domain adaptation method confronts the same challenge of fragmentation facing other topic-based adaptation approaches. Iyer et al. (1994) and Iyer and Ostendorf (1999) point out that a too aggressive partitioning of the training data may aggravate data spareness issues for the component language models, which are trained on the individual partitions. In order to address this issue, they interpolate the topic-specific models with a general model trained on the entire data set,

$$P(s) = \sum_k \lambda_k \prod_i [\alpha_i P_k(w_i|h(w_i)) + (1-\alpha_i) P_g(w_i|h(w_i))], \qquad (2)$$

where P_k is a topic model and P_g a general model.

The key insight of our paper is that explicit interpolation as in Eq. (2) is not necessary for RNNLMs. Rather, we can make use of a well-planned curriculum that exploits the fact that training data presented later in the training process contribute more to the final state of the RNNLM.

We plan this curriculum so that it starts with general data, corresponding to the overall collection, and moves to specific data, corresponding to the individual sub-domains. Sparse data issues are alleviated by allowing all available training data to contribute to the component model corresponding to each individual sub-domain. The arrangement of the ordering of the training data implicitly adjusts the weights between the general model and component models. Because curriculum learning provides this possibility for implicit interpolation, explicit interpolation, as in Eq. (2), is not needed in order to train sub-domain-adapted RNNLMs. As previously mentioned, because the curricula make use of all available training data, only changing the order of presentation, our sub-domain-adapted RNNLMs are able to avoid the dangers of fragmentation. In the next section, we go on to discuss curriculum learning in more detail.

2.2. Curriculum Learning for Neural Networks

An investigation at the intersection of cognitive science and machine learning carried out by (Elman, 1993) is credited with laying the groundwork for curriculum learning. The goal of this work was to understand the contributions of restrictions existing early in the learning process to the final success of learning.

The topic was taken up again in the machine learning community by Bengio et al. (2009) under the label 'curriculum learning'. Here, the interest was not generally on restrictions during the learning process, but rather on restrictions on the input data. Specifically, Bengio et al. (2009) shows that final ability improves if

the learner is first presented with simple input and then moved to more complex input. Curriculum learning is connected to many other techniques used for learning, such as boosting and transfer learning. Bengio et al. (2009) characterizes the special emphasis of curriculum learning as guiding the optimization process, especially in a way that steers it towards better local minima. The discussion in Bengio et al. (2009) opens some intriguing vistas of inquiry. First, the notion that curriculum learning 'guides the optimization process' contributes a valuable insight into why curriculum learning works, but a more concrete understanding could be useful for applying curriculum learning in practice. Second, the notions of 'simple' and 'complex' are very general, and have multiple useful interpretations. Bengio et al. (2009) provides a statistical formalization of a curriculum that moves from 'simple' to 'complex' based on the idea that the entropy of the data should increase so that the diversity of the training data increases. However, the experiments also show that naively creating two classes of 'simple' and 'complex' samples already provides improvement.

Our work is based on the idea that in language model adaptation scenarios, 'simple' can be productively equated with 'general' and 'complex' can be equated with 'specific'. We take the difference between 'general' and 'specific' to be related to differences of both word choice and word usage, which are in term reflected in different n-gram distributions. However, in application scenarios it will not be practical to predict what this differences would be, or, expect to be able to calculate them precisely. For this reason, we keep our definition of 'general' vs. 'specific' naive, and assume nothing more than that the difference reflects the variations within a particular domain that lead to that domain being considered heterogeneous.

This assumption is consistent with the discussion in Elman (1993) on the reasons for which a language model is able to benefit from a particular ordering of training data to improve its ability for predicting words. According to (Elman, 1993), in the first learning stage the network learns a functional representation scheme that encodes an underlying structure of the data. Then, in the second learning stage, the learner moves to learn the surface reflexes of this underlying structure. Based on the initial 'simple' examples, the learner is able to create an approximation of the solution space that constrains the learner from making 'mistakes' when it is presented with more complex data. In our case, we assume that by presenting the RNNLM first with 'general' input, it will learn overall patterns in the language, and that these will allow it to derive maximum benefit from the 'specific' input for the sub-domain, since it will be be protected from over-fitting as it adapts to the sub-domain.

The application of curriculum learning to RNNLMs proposed in (Mikolov et al., 2011b) can be considered a progenitor of our work. (Mikolov et al., 2011b) interprets 'simple' to mean 'closest to the target data'. Results are reported on two experiments in which the data presented to the RNNLM ordered by increasing perplexity with respect to the development data. The first experiment demonstrates that data ordering can lower the convergence time. The second experiment demonstrates that a combination of data ordering and filtering of high perplexity data leads to a reduction of the perplexity of the resulting RNNLM measured on the evaluation data. Considering the second experiment as a proof-of-concept, our work explores in depth the application of curriculum learning to RNNLM adaptation. Specifically, we establish that the benefits of curriculum learning go beyond an optimization of the training set to be useful to address two further challenges

facing language models for speech recognition: *within-domain adaptation* and *limited-data within-domain adaptation.*

This paper builds on and extends our previous work (Shi et al., 2013), in which we first introduced the idea of applying curriculum learning to adapt RNNLMs. This work provided an initial exploration of curriculum learning for within-domain adaptation. Here, we examine the issue of within-domain adaptation in greater depth and detail and also extend the idea of curriculum learning for RNNLMs to limited-data within-domain adaptation.

2.3. Recurrent Neural Network Language Models

The pre-cursor of the today's RNNLMs can be considered to be (Bengio et al., 2003), which proposed feed-forward neural network language models. In these models, each word in the vocabulary is mapped by a shared parameter matrix to a real vector. Mikolov et al. (2010, 2011c) further extended the feed-forward neural network language model to a recurrent neural network by incorporating history information into the input layer. In his RNNLM, the input layer consists of the input word and also the activated hidden layer associated with the previous word.

As shown in (Mikolov et al., 2011a), RNNLMs outperform other advanced language models currently in common use. As previously mentioned, the superior performance of RNNLMs can be attributed to two properties. First, RNNLM projects each word from a large discrete vocabulary space to a small continuous vector space. The mapping from the discrete to the continuous space enables the model to learn generalization. The continuous representation of words can be effectively used to capture the similarities between different words and reduce the effect from data sparseness (Mikolov et al., 2013). Second, the recurrent structure of RNNLMs equips them with a compact memory. Theoretically, recurrent neural

networks can store relevant information from previous time steps for an arbitrarily long period of time, making it possible to learn long-term dependencies.

Curriculum learning can be applied to different types of neural network language models, however, in this paper, we only focus on applying curriculum learning to RNNLMs. Our choice is informed by the documented state-of-the-art results achieved by RNNLMs, as demonstrated on several benchmark data sets by RNNLM Mikolov (2012).

Recently, Mikolov and Zweig (2012) proposed a context dependent RNNLM, in which the context information is a latent topic vector obtained from the preceding text using LDA. In this paper, we also use LDA to create topical representations. However, instead of integrating latent topic information into one general model, we use it in order to produce a clustering of the data that serves as the basis for producing a series of sub-domain-adapted RNNLMs, which are then applied in combination.

3. Curriculum Learning For RNNLMs

In this section, we introduce three different methods of applying curriculum learning in constructing sub-domain-adapted models. At the beginning of the section, we describe the basic structure of a RNNLM. Then we give the details about the curriculum learning methods used in this paper.

3.1. Recurrent Neural Network Language Models

The recurrent neural network language models adopted in our work originated from (Mikolov et al., 2010). As shown in Fig. 1 (solid line parts represent conventional RNNLM), it has three layers: an input layer x, a hidden layer h and an output layer y. It is characterized by the loop between the input layer and hidden

layer, which acts as a short abstract memory that stores previous information. The input vector x_t at each time t consists of the current word encoded as a one-hot vector w_t as well as a copy h_{t-1} of the previous hidden neurons. The output layer y_t is factorized to two parts: one is the word part w_{t+1}, the other one is the class part c_{t+1}. The class is determined on the basis of the frequencies of the words in the training data. Classes are used in RNNLMs to speed up training. Instead of updating the whole weight matrix connecting the hidden layer and the output layer, training in the case of an RNNLM that makes use of classes need only to update two relatively small weight matrices. One matrix connects the hidden layer to the class section of the output layer. The other one connects the hidden layer to the words in the output layer that belong to the same class as word w_{t+1}. The sigmoid function and softmax function are used as the activation functions in the hidden layer and output layer, respectively. The weight matrix between the input layer and hidden layer is estimated by backpropagation-through-time (BPTT)(Mikolov et al., 2011c), which actually unfolds the loop as the deep neural network. In this paper, we set the BPTT block size to 10 and BPTT times to 4. Basically, using such settings, the weight matrix connecting input layer to the hidden layer is updated only each time the training has processed 10 words. Each BPTT update unfolds the loop as a deep neural network with 13 (BPTT block size + BPTT times -1) hidden layers.

3.2. Three Curriculum Learning Methods

In this paper, we propose three different curriculum learning strategies for the training of sub-domain-adapted RNNLMs. The commonality of the three strategies is that they move from presenting the RNNLM with general data to presenting it with specific data. The strategies progressively give specific patterns in the data

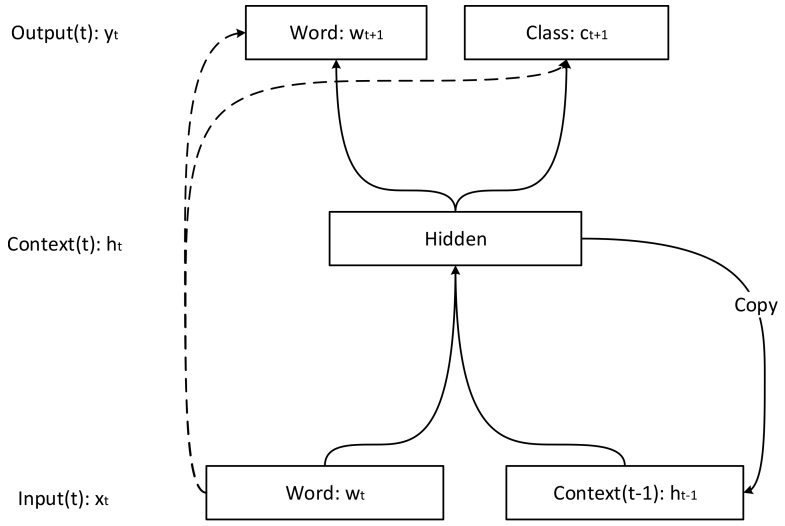

Figure 1: Recurrent Neural Network Language Models with Maximum Entropy Extension. Dash line parts represent the Maximum Entropy Extension.

more influence during model training.

Start-from-Vocabulary (*Start-Vocab*) This curriculum uses the entire training data set to construct the vocabulary of the sub-domain-adapted RNNLM, but then only uses the training data from the corresponding sub-domain for training.

Data sorting (*Data-Sort*) This curriculum uses the entire data set to train the sub-domain-adapted RNNLM. The data is sorted such that the final data presented during training is the data from the target sub-domain. The validation data is also selected from the corresponding sub-domain. The learning rate is halved when the model can not achieve improvement on the validation data. Model training is terminated when continuous learning rate reduction no longer achieves improvement on the validation data.

All-then-Specific (*All-Specific*) Each sub-domain-adapted model starts with a number of epochs training on the whole data and then learns from its specific sub-domain data. In the general training period, we choose validation data that covers all the sub-domains. Training in the general period is controlled by halving the learning rate when the model cannot achieve improvement on the general validation data. In the specific training period, the validation data is selected from its corresponding sub-domain. Training in the specific phase is terminated when continuous learning rate reduction cannot achieve improvement over the validation data set of the specific sub-domain. In the specific training, the learning rate starts from the same learning rate when the general training ends.

We now briefly discuss the rationale for these models and the difference between the three strategies. Of the three, the *Start-Vocab* method uses the least general pattern information. The resulting sub-domain-adapted RNNLM model has only been exposed to patterns from its target sub-domain during the training process. The portion of the weight matrix that to words that occur in the overall training set, but not in the sub-domain training data, is random initialized, but not updated further during training. The effect is that the words that are in the sub-domain get small probabilities. Both the *Data-Sort* and the *All-Specific* strategies use the whole training data including the sub-domain data. The strategies differ with respect to the point at which they move from general patterns to specific patterns. *Data-Sort* makes the transition inside each epoch, but *All-Specific* makes the transition outside of the epochs. Under the *Data-Sort* strategy, the model is shaped to approach a domain-specific optimum within each training epoch. Under the *All-Specific* strategy, the model is first fully optimized with respect to general

patterns, and then pulled from this point to a domain-specific optimum.

3.3. Experimental Set-Up

In this section, we explain the design of our experiments and provide the details of the implementation of the experimental framework that we used.

3.4. Evaluation

Our experiments compare our proposed curriculum learning methods with two baselines. The first, is the unadapted baseline ('base'). This baseline is an RNNLM that has been trained on the entire data set in its natural order. The second is conventional linear interpolation ('int'). The 'int' models are sentence level mixtures of individual sub-domain-adapted models, cf. Eq (2). Each sub-domain-adapted model is a linear interpolation of a general model with a specific model trained only using the data in the training set belonging to its specific sub-domain.

We use several metrics to report the results of our experiments. The perplexity (PPL) is a commonly used metric for measuring language models. It is calculated as the geometric average of the inverse probability of the words on the test data.

$$PPL = (\prod_{i=1}^{t} P(w_i|h(w_i)))^{-\frac{1}{t}}, \tag{3}$$

where $h(w_i) = w_1 w_2 ... w_{i-1}$. Word prediction accuracy WPA (van den Bosch, 2006) is a measure of the performance of language models in practice.

$$WPA = \frac{C}{N}, \tag{4}$$

where C is the number of correctly predicted words and N the number of total words. Many applications in the area of natural language process, such as spell checking and sentence completion, use WPA. Word Error Rate (WER) is used for

those experiments for which we carry out N-best re-scoring, described in more detail below. Based on a speech recognition system, using minimum edit distance, the prediction sentence is compared with reference sentence to get the number of substitutions S, the number of deletions D and the number of insertions I.

$$WER = \frac{S + D + I}{N}, \tag{5}$$

where N is the number of total words.

3.5. RNNLM *Framework*

In the experiments, the language models are applied to two closely related, but different tasks, word prediction and N-best rescoring. In word prediction, a word that gets maximum probability in the output layer is chosen as the predicted word. In N-best rescoring, the weighted combination of acoustic model score, language model score and word insertion penalty is used to select one best hypothesis sentence. The weight of the combination is determined by a held-out development N-best list data set through grid search.

Because of the differences between the different strategies for curriculum learning, different stopping criteria for training are applied. Start-from-Vocabulary (*Start-Vocab*) and Data sorting (*Data-Sort*) stop when the sub-domain-adapted model can not achieve additional PPL improvement on the validation data. In each case, the validation data set is selected from the sub-domain, but mutually is exclusive with the training and test sets. *All-Specific* training consists of two phases, one for whole data training, the other for sub-domain data training. In all data training, we specify the number of training epochs. The sub-domain training starts when the all data training finishes the specified number of epochs. The sub-domain training stops when the model can not achieve additional improvement on

the sub-domain validation data.

The experiments are carried out using the RNNLM toolkit [1]. In some experiments, we apply the Maximum Entropy Extension to RNNLMs, which uses a weight matrix that directly connects input features to the output layer. The structure of the Maximum Entropy Extension of RNNLMs are shown in Fig 1. The dash line in the figure represents the Maximum Entropy Extension part of RNNLMs. When the extension is applied to a condition it is applied to all experiments carried out for that condition and also indicated clearly in the text. By applying this extension, we are able to report the state of the art optimal values that are reachable by RNNLMs on our data.

4. Within-domain Language Model Adaptation

This section presents our investigation of the use of curriculum learning to address the challenge of *within-domain adaptation* for RNNLMs. Table 1 presents an overview of the experiments. As shown in table, the experiments start from the situation that the sub-domain information is known during both training and testing, move to the situation that the sub-domain information is only known during training, and end with the situation that sub-domain information is unknown during both training and testing. Condition 1, in which sub-domain information is known during testing, is an oracle condition that lets us test the sub-domain-adapted RNNLMs individually on their domains, and gives us insight into their power to predict sub-domains. Condition 2 and 3 address the real-world case in which the sub-domain labels of the test data are unknown. In these cases, we use

[1] http://www.fit.vutbr.cz/ imikolov/rnnlm/

combinations the individual sub-domain-adapted RNNLMs. We experiment with a *hard decision combination* and a *soft decision combination*, described in more detail below.

Table 1: Within-domain adaptation experiments (Section 4): Overview of experimental conditions.

	Condition 1	Condition 2	Condition 3
Training data sub-domain	known	known	unknown
Test data sub-domain	known	unknown	unknown
Language Model	sub-dom.-adapted RNNLM	combinations of sub-dom-adapted RNNLMs	combinations of sub-dom-adapted RNNLMs

4.1. Sub-domain Information Known for Training Set

In this section, we investigate the case in which sub-domain information is known for the training data. Such situations occur naturally in cases that the sub-domain information can be obtained at the time that the data is collected. For example, if a sub-domain is associated with a certain phone, microphone or known-location. In this situation, sub-domain-adapted RNNLMs are trained on sub-domain-labeled data. We first present the data set and then go on to present the results of our experiments on Conditions 1 and 2.

4.1.1. Sub-domain Known Data Set: Spoken Dutch Corpus (Conditions 1 and 2)

For our investigation of cases involving known sub-domains, we choose the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) (Oostdijk et al.,

2002; Oostdijk, 1999). This corpus is representative of a heterogenous data set consisting of data that has been recorded in multiple social-situational settings. In previous work, we investigated the characteristics of the CGN corpus (Shi et al., 2011b). This work revealed that different socio-situational settings are characterized by different word distributions and different syntactic constructions.

Table 2 presents a detailed description of the CGN data set. The data set contains audio recordings of standard Dutch spoken by adults in Netherlands and Flanders. It comprises nearly 9 million words divided into 14 sub-domains that correspond to different socio-situational settings. The data sets contain recordings ranging from read speech to lectures and including spontaneous conversations. *comp-i* to *comp-o* involve a single speaker and *comp-a* to *comp-h* contain dialogues or multilogues. In the table, the column labeled "Tokens" gives the number of running words in each sub-domain.

From the CGN data, we randomly selected 80% for training, 10% for validation and 10% for testing. In the test data, those words that are not in the language model vocabulary are replaced by an out-of-vocabulary token (the OOV rate is 3.8%). All RNNLMs trained on CGN data have a hidden layer of 300 neurons, and use 100 classes and 4 times BPTT with a block size of 10.

4.1.2. Condition 1: Sub-domain Known for Training and Test Set

Our investigation of the oracle condition, i.e., sub-domain information is known at training and testing time, allows us to understand the improvements that sub-domain-adapted RNNLMs achieve over our baseline, an unadapted RNNLM ('base') and over adapted models created using conventional linear interpolation ('int')(Eq (2)). The experiments reveal the relative improvements that sub-domain-adapted RNNLMs are able to achieve as well as their predictive power for sub-domains.

Table 2: Overview of the Spoken Dutch Corpus (CGN).

Label	Socio-situational setting	Tokens
comp-a	Spontaneous conversations ('face-to-face')	2,626,172
comp-b	Interviews with teachers of Dutch	565,433
comp-c&d	Spontaneous telephone dialogues	2,062,004
comp-e	Simulated business negotiations	136,461
comp-f	Interviews/ discussions/debates	790,269
comp-g	(political) Discussions/debates/ meetings	360,328
comp-h	Lessons recorded in the classroom	405,409
comp-i	Live (eg sports) commentaries (broadcast)	208,399
comp-j	Newsreports/reportages (broadcast)	186,072
comp-k	News (broadcast)	368,153
comp-l	Commentaries/columns/reviews (broadcast)	145,553
comp-m	Ceremonious speeches/sermons	18,075
comp-n	Lectures/seminars	140,901
comp-o	Read speech	903,043

The results of the experiments are reported Table 3 (perplexity) and 4 (WPA). These results were produced by testing each sub-domain-adapted RNNLM (corresponding to each of the three curriculum learning strategies *Start-Vocab*, *Data-Sort*, and *All-Specific*) on its corresponding sub-domain. *Start-Vocab* and *Data-Sort* were trained for until the convergence criteria were met. *All-Specific* was trained with general data for 10 epochs and further with sub-domain specific data until the convergence criterion was met.

The results in Table 3 and 4 allow us to make several important observations.

Table 3: The perplexity (PPL) comparison of conventional general RNNLM (base), sub-domain-adapted models in sentence level mixture models using linear interpolation (int) and sub-domain-adapted models using different strategies of curriculum learning on CGN data set under the oracle situation. *Start-Vocab* is Start-from-Vocabulary, *Data-Sort* is Data-Sorting, and *All-Specific* is All-then-Specific. The sub-domains listed in the table are those for which the sub-domain-adapted models achieve substantial improvement. The bottom row of the table gives the *average* of the perplexities that are achieved by 14 different sub-domain-adapted models tested on their specific sub-domains.

comp	base	int	*Start-Vocab*	*Data-Sort*	*All-Specific*
e	80.1	52.3	62.2	47.6	47.8
g	283.4	190.6	245.2	180.0	179.4
i	341.3	141.5	189.8	146.9	145.8
k	553.1	222.4	292.9	221.6	230.3
o	480.2	274.1	328.4	269.2	261.3
average	238.7	168.4	231.2	154.0	152.9

Table 4: The percent word prediction accuracy (WPA) comparison of conventional general RNNLM (base), sub-domain-adapted models in sentence level mixture models using linear interpolation (int), and sub-domain-adapted models using different strategies of curriculum training on the CGN data set under the oracle situation. *Start-Vocab* is Start-from-Vocabulary, *Data-Sort* is Data-Sorting, and *All-Specific* is All-then-Specific. The sub-domains listed in the table are those for which the sub-domain-adapted models achieve substantial improvement.

comp	base	int	*Start-Vocab*	*Data-Sort*	*All-Specific*
e	24.5	25.0	23.7	26.6	25.9
g	15.9	16.5	16.0	17.6	17.7
i	16.5	19.9	18.8	19.7	19.8
k	14.5	20.0	19.7	19.9	19.8
o	14.2	15.6	15.2	16.3	16.4
total	20.6	21.3	20.0	23.0	23.3

First, consulting the total WPA at the bottom of Table 4, we can see that baseline model (base), which is a RNNLM that has been trained on the whole data set without using curriculum learning, is outperformed by both conventional linear interpolation (int) and also the sub-domain-adapted RNNLMs trained with the *All-Specific* or *Data-Sort* strategies for curriculum learning. This result confirms that language models are very sensitive to differences in sub-domain, and illustrates the importance of taking sub-domain information into consideration. In some cases the benefits of adapting to sub-domains are particularly dramatic. For example, for "News" (comp.-k), all the adapted models achieve an over 50% reduction in terms of perplexity and more than 30% of improvement in terms of WPA.

Second, the results in Table 3 and 4 demonstrate that overall the *All-Specific* and *Data-Sort* curriculum learning strategies outperform conventional linear interpolation. As previously mentioned, a priori, the implicit interpolation carried out by curriculum learning is superior to conventional linear interpolation. Because curriculum learning achieves a balance between general and specific data without requiring weights to be tuned on the validation data, as is the case for conventional linear interpolation. The results provide evidence that in addition to being easier to apply in practice, curriculum learning also delivers a performance improvement over conventional linear interpolation.

Third, these experimental results reveal that among the three curriculum learning strategies, *Start-Vocab* performs the worst consistently over sub-domains. Recall that *Start-Vocab* adopts only the vocabulary of the full training data set, and otherwise trains each sub-domain-adapted RNNLM only on sub-domain specific data. Apparently, *Start-Vocab* does not offer enough exposure to general data dur-

ing the training process and sub-domain data are insufficient to allow the RNNLM to generalize robustly. Because *Start-Vocab* exploits only data from a single sub-domain, low performance of *Start-Vocab* can be considered as a result of the same data fragmentation problem that was mentioned above, i.e., a single topic-specific domain can easily fail to contain enough data to train a topic-specific model.

Table 5: The average of the perplexities (PPL) and the overall word prediction accuracies (WPA) comparison using *All-Specific* curriculum learning with increased number of epochs of general data training (gen.).

	2 epoches	4 epoches	6 epoches	8 epoches	10 epoches
average PPL	170.3	163.2	154.9	153.5	152.9
total WPA	22.3	22.9	23.2	23.3	23.3

Fourth, in Table 3 and 4 it can be seen that the performance of sub-domain-adapted RNNLMs using the *Data-Sort* curriculum learning is similar to the performance of those trained with the *All-Specific* strategy. Recall that in each epoch of *Data-Sort* training, the data is sorted, meaning that within each epoch the RNNLM is first trained by the general data and then further trained by the specific sub-domain data. In contrast, *All-Specific* is trained using a certain number of epochs of the general data set followed by an additional number of epochs of specific sub-domain until the convergence criterion is met. Table 5 presents results showing the impact of the number of general-data epochs used before beginning specific-data epochs. It can be seen that the maximum performance is reached with 10 epochs, i.e., the largest number of general training epochs. It should be noted that 10 epochs represents nearly a completely trained general model, which would other-

wise reach its convergence criterion after 12 training epochs. This result confirms the importance of the contribution of the general data to training the sub-domain adapted RNNLMs.

To complete the discussion of our approach in the case of known sub-domain information, we would like to comment on its generality. The approach of applying curriculum learning to adapt RNNLMs can be applied to improve the performance of any RNNLM. In other words, it can anticipate that as innovations improve the underlying RNNLM, our approach will continue to deliver benefits on top of the underlying gains. In order to illustrate this point, we turn to a set of advanced RNNLMs that incorporate additional features, whose effectiveness we have demonstrated in previous work Shi et al. (2012b). Experimental results on these models are reported in Table 6. Here, "RNNLM + POS" designates and RNNLM combined with part-of-speech (POS) features. "RNNLM +sub-domain" uses the sub-domain information differently from how it is used in this paper. It treats the sub-domain information as an additional feature in the input layer for RNNLMs. In the table, "RNNLM +complete" means combining RNNLM with POS, sub-domain and lemma. As shown in Table 6, applying *All-Specific* and *Data-Sort* to RNNLMs that use syntactic and linguistic features can achieve additional improvement, in the situation that all these features are known during testing. By comparing "RNNLM +sub-domain" with "RNNLM +*Data-Sort*" and "RNNLM +*All-Specific*", we also find that curriculum learning is a more effective way of using sub-domain information than the technique applied in Shi et al. (2012b).

To summarize, this section has shown the importance of adaptation in general, and also illustrated the superiority of curriculum learning for language model adaptation with respect to conventional interpolation. We have also shown that the

Table 6: Combination of curriculum learning with other improved RNNLMs that use syntactic and linguistic features.

models	WPA (%)
base	20.6
RNNLM +POS	22.6
RNNLM +sub-domain	20.8
RNNLM +lemma	21.6
RNNLM +complete	22.9
RNNLM +*Start-Vocab*	20.0
RNNLM +*Data-Sort*	23.0
RNNLM +*Data-Sort* +POS	23.7
RNNLM +*Data-Sort* +lemma	23.2
RNNLM +*All-Specific*	23.3
RNNLM +*All-Specific* +POS	**24.1**
RNNLM +*All-Specific* +lemma	23.4

approach is general in that sense that the curriculum learning adaptation strategy has the ability improvement in the underling RNNLM into an overall improvement. In the next section, we move on to investigate whether the performance improvements offered by curriculum learning can be extended to a condition in which the sub-domain of the test data is not known.

4.2. Experiment 2: Sub-domain Known for Training Set and Unknown for Test Set

We now turn to the case in which sub-domain information is known for the training data, but not for the test set. In this case, we do not know in advance which sub-domain-adapted language model should be applied to a given segment of speech, which in the case of the CGN data set, is a sentence. For this reason, instead of applying a single sub-domain-adapted language model, we make use of approaches that combine sub-domain-adapted language models.

The first approach makes a *soft decision* about the correct sub-domain of the input sentence using a linear combination of the scores generated for a given sentence by all of the sub-domain-adapted language models. We use the conventional form for a sentence-level mixture, given above in Eq (1). In our implementation, the interpolation weights λ_k are dynamically determined according to the following heuristic method:

$$\lambda_k(s) = \frac{p_k(s, h_s)}{\sum_i p_i(s, h_s)}, \tag{6}$$

where $p_k(s, h_s)$ is the joint probability from the kth sub-domain-adapted for the present sentence s and its history h_s. Basically, the sub-domain-adapted model that assigns higher probability for current sentence receives more weight to determine the next word.

The second approach makes a *hard decision* for each input sentence. Our previous work has examined the relative advantages of the hard decision vs. the soft decision for other types of language models, and revealed it to be important (Shi et al., 2012a, 2011a). Here, we investigate both with respect to the combination of RNNLMs. In order to motivate the use of the hard decision to combine sub-domain adapted RNNLMs, we carried out an analysis of the ability of RNNLMs to predict the identity of sub-domains. Sub-domain-adapted RNNLMs can be used to predict the sub-domain of a sentence s $C(s)$ as follows:

$$C(s) = \arg\max_k p_k(s, h_s), \tag{7}$$

where h_s is the history of sentence s. $p_k(s, h_s)$ is the joint probability of sentence s with its $history$, which is assigned by the kth sub-domain-adapted model. Our experiments with sub-domain prediction yielded very good results. For nine of the thirteen sub-domains in the CGN data set, the prediction accuracy was above 95%. This result suggests that the hard decision method for combining sub-domain-adapted RNNLMs could possibly approach the neighborhood of the performance observed in Section 4.1.2 under the oracle condition where the identity of the sub-domain is known for the test data.

Table 7 shows perplexity and word prediction accuracy results of soft and hard decision on CGN data, in which sub-domain context is not available to the models during the testing. The soft decision results show that the soft decision combination of the sub-domain-adapted models using interpolation achieves the lowest perplexity. However, this performance does not transfer to the word prediction accuracy. Using soft decision combination method, *All-Specific* with 10 epochs of general training obtains the best WPA. Based on sentence level WPA, "*All-Specific*-hard" achieves significant improvement over baseline model. Although

Table 7: Perplexity and word prediction accuracy result of the Spoken Dutch Corpus (CGN). The sub-domain of the test data is unknown.

models	PPL	WPA (%)
base	114.8	20.6
int-soft	98.1	20.7
Start-Vocab-soft	118.9	19.8
Data-Sort-soft	103.4	21.8
All-Specific-soft	104.1	22.7
int-hard	-	21.3
Start-Vocab-hard	-	20.1
Data-Sort-hard	-	22.1
All-Specific-hard	-	23.2

"*Data-Sort*-soft" also achieves substantial improvement over baseline model, it did not achieve statistical significant improvement.

Comparing the soft and hard combinations, we find that the hard combination performs better than soft combination method in terms of WPA. In addition, the hard combination method is also computationally less complex than soft combination one in word prediction task. Using the soft decision method for word prediction, the system needs to put the hidden layer and output layer of each sub-domain-adapted RNNLM into memory. In addition, with the soft decision method, all the output layers of sub-domain-adapted RNNLMs have to be linear interpolated together to do prediction. One probable reason why hard decision performs better than soft decision is that using Eq (7) has good sub-domain prediction capability that achieves over 95% accuracy in nine of the thirteen sub-domains in the CGN

data set. However, the main disadvantage of the hard decision method is that it does not produce a normalized probability. Because during testing, probabilities of different sentences are assigned by different models.

4.3. Experiment 3: Sub-domain Unknown in both Training and Test Sets

We now turn to the case in which sub-domain information is known for neither the training nor the test set. We are interested in exploring the abilities of curriculum learning in cases in which sub-domain information is not collected at the time that the data is captured, but instead must be inferred. In this section, we present an additional experiment about using the proposed RNNLM adaptation method to a benchmark test data set which is Wall Street Journal (WSJ). WSJ data set is a well-known data set that has been used in previous work by Mikolov et al. (2010); Mikolov and Zweig (2012). In previous sections, it shows that RNNLMs adaptation using curriculum learning can outperform both the conventional RNNLM and the sentence level mixture models on CGN data. However, we notice that the CGN covers many speech sub-domains. Each of these sub-domains actually is dramatically different with other sub-domains in style. WSJ is about newspaper articles. We can assume that these newspaper articles belong to different topics, making it a reasonable choice to investigate topical structure. However, in WSJ the style is same, providing an interesting contrast with CGN. In this section, we will investigate the speech recognition performance of the proposed method for WSJ data set using latent topic clustering for the training data.

4.3.1. WSJ *Data*

In WSJ, we use 100-best speech recognition list from the DARPA WSJ'92 and WSJ'93 test data sets, as used by Mikolov et al. (2010); Wang and Harper (2002).

In the 100-best list set, 333 sentences are used as development data for tuning the interpolation of language models score and acoustic model score (DEV). The rest, 465 sentences, are used for evaluation (EVAL). The oracle WER of 100-best lists for development data and evaluation data are 6.1% and 9.5%, respectively. The training corpus, referred to as the NYT data set, contains 37M words of running text from the New York Times section of English Gigaword. The validation data set contains 186K words. A held-out set of 230K words is used for testing (TEST). The vocabulary size is 194K.

4.3.2. Experiments

We use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to generate the latent topics for the training data. LDA is a probabilistic generative model, which use bag-of-words strategy to assign each document with latent topic representation. In this paper, each document is a sentence. Basically, we generate the latent topic representation for each sentence in the training data. In the training data, each sentence is treated as a document. The topic distribution can be viewed as a reduced latent topic representation for each sentence. Based on this normalized continuous representation of each sentence, we then use the k-means clustering method to partition the training data.

In the N-best list rescoring experiment, all RNNLMs have a hidden layer of 200 neurons, and use 100 classes and 4 times BPTT with a block size of 10. Additionally, we make use of the Maximum Entropy extension in the form of 1 billion elements that directly connect the input features to output layers using maximum entropy extension.

Table 8 shows the word error rate performance of the proposed RNNLM adaptation method in N-best rescoring. The models using *Data-Sort* curriculum learning

achieves 0.2% improvement over the conventional RNNLM in terms of word error rate. Although this improvement is too small to be significant (based on the evaluation N-best list with 465 sentences, a paired t-test shows that $mean = 0.025\%$, $t = 0.146$, $p = 0.886$), we note that the gains we obtained here are comparable in magnitude to those reported on another type of RNNLM extension, namely context based RNNLM Mikolov and Zweig (2012).

Compared with the experimental results on the CGN data, on the WSJ data set, the RNNLM adaptation using curriculum learning makes only small gains over the conventional RNNLM. This relatively smaller gains are unsurprising in view of the type of data. The WSJ data set, in contrast with the CGN data, is homogenous with respect to style (i.e., it is all news data). As a result, applying clustering to training data generates sub-domains, but these sub-domains cannot be expected to be strikingly distinct. If the training and the test data were well matched with respect to topics, then we could expect the model to exploit topical structure. However, the training data (NYT section of English Gigaword) and the test data were collected during two different time windows, and as such topics discovered in the training data are not necessarily useful for test. An important insight is that the exploitation of curriculum learning to adaptation demonstrates a sensitivity to the difference between the general domain and the sub-domains used for adaptation.

5. Limited-data Within-Domain Adaptation

Next we turn to our second set of experiments, in which we investigate the ability of curriculum learning to adapt language models in the face of the *limited-data within-domain adaptation* challenge. Limited-data domain adaptation is necessary in many real-world scenarios, since language models that are used in prac-

Table 8: The word error rate (WER) comparison on WSJ data set. 'base' is the conventional RNNLM. 'kn-5' is a Keneser-Ney 5-gram language model. 'int' is a conventional linear interpolation of sentence level mixture models (Eq. (1,2)). "*Start-Vocab*", "*Data-Sort*" and "*All-Specific*" are RNNLM adaptation using different types of curriculum learning. "T" column represent the number of topics. "hard" and "soft" represent hard decision (Eq. (7)) and soft decision (Eq. (1) and (6)), respectively.

models	T	dev (%)	eval (%)	T	dev (%)	eval (%)
kn-5		12.1	17.3			
base		10.3	14.9			
base-int-soft	5	11.2	15.3	10	11.3	15.9
base-int-hard	5	11.5	15.8	10	11.2	16.2
Start-Vocab-soft	5	11.6	16.1	10	11.5	16.3
Start-Vocab-hard	5	11.3	16.3	10	11.5	16.5
Data-Sort-soft	5	10.2	14.7	10	10.3	15.2
Data-Sort-hard	5	10.4	14.7	10	10.2	15.2
All-Specific-soft	5	10.3	15.1	10	10.4	15.0
All-Specific-hard	5	10.1	15.1	10	10.3	14.9

tical applications generally need frequent updating. In many cases, updates are carried out by retraining language models from scratch, adding a small amount of new, yet-unseen training data to the original large existing training data set.

5.1. Experimental data set

Our choice of data sets is intended to emulate the real-world situation, in which the system has already been trained on a relatively large amount of existing data and a smaller amount of yet-unseen data becomes available. In this experiment,

the large existing data set that we choose is the NYT section of English Gigaword (see Section 3.5.3), which we used in previous section. The yet-unseen data set is the Penn Treebank (PTB) text data set. We use section 00-20 for training ($972K$ word tokens), section 21-22 for validation ($77K$ word tokens) and section 23-24 for testing ($84K$ word tokens).

5.2. Experiments

In the experiment, we use the *All-Specific* curriculum learning method for limited-data domain adaptation. Because the *All-Specific* curriculum learning method trains existing language model as new data becomes available, it is uniquely suited to address the limited-data within-domain challenge. Note that, in contrast, neither *Start-Vocab* nor *Data-Sort* methods is fitting to use for limited-data domain adaptation. *Start-Vocab* is not suitable since it requires the vocabulary to be calculated on the entirety of the training data including adaptation training data. *Data-Sort* is not also suitable because it requires the entirety of the training data be presented to the RNNLM in every training epoch. Under the definition of the limited-data within-domain challenge, the adaptation data are not available to the system at the moment in which the original model is trained.

In the experiment, we compare the *All-Specific* curriculum learning method with a baseline that uses conventional linear interpolation to combine the existing language model with a new language model trained on the yet-unseen data. The models produced by conventional linear interpolation and by *All-Specific* curriculum learning both maintain the same vocabulary as the existing language model.

In our experiment, the vocabulary of the existing language model, determined by the NYT, contains $194k$ words. All the words in the yet-unseen data that are not in the vocabulary are treated as OOV words. In the yet-unseen data set, the OOV

rate for training data is 9.3%, for validation data 7.1% and for testing data 7.9%. During testing, the probability of OOV words is set to 10^{-8}. All the models in this subsection are trained with the same settings as models from subsection 4.3.

In order to cleanly isolate the source of the contribution of our approach, we include an additional baseline in which the training data is randomly shuffled in each epoch. The top part of Table 9 shows that this shuffling does not have a substantial influence to PPL and WPA. We note that an advantage of data randomly shuffling the training data the training process can be sped up—we observed that nearly 20% fewer training epochs were used.

Table 9 shows the existing language model can achieve substantial improvement on PTB testing data in terms of perplexity and WPA by specific training on the PTB training data. If only 20% of the PTB training data is used, using *All-Specific* curriculum learning the existing language models can achieve 37.7% reduction in terms of PPL and 11.5% relative improvement in terms of WPA. The performance of the language models can be improved by using more PTB data in the training process. As shown in the table, curriculum learning applied for limited-data domain adaptation can produce RNNLMs that are better, in terms of WPA, than conventional RNNLMs trained only on PTB data.

Table 9 also shows that the model trained by *All-Specific* curriculum learning using 80% of yet-unseen data achieves similar performance as that achieved by a model that uses the interpolation of the existing language model and a language model trained on yet-unseen data based on the same vocabulary as existing model. This result suggests that curriculum learning can be used as an effective method to implicitly weight the patterns in the existing training and the pattern in the yet-unseen training data. The results of only using 20% yet-unseen train-

Table 9: Comparison of RNNLMs trained by only yet-unseen data (PTB base), trained on existing data (NYT base), linear interpolation of RNNLMs trained on existing and yet-unseen data (int NYT +PTB), and using the *All-Specific* curriculum learning method (*All-Specific* CL NYT +PTB). Results are reported on yet-unseen test data. The percentage of yet-unseen data used in training is given in the column marked 'ratio'. 'PTB rand' and 'NYT rand' means using training data random shuffling.

models	ratio	PPL	WPA (%)
PTB base	1.0	125.9	24.6
PTB rand	1.0	124.4	24.6
NYT base	0.0	180.7	21.7
NYT rand	0.0	181.4	21.7
int NYT + PTB	0.2	121.5	23.7
int NYT + PTB	0.4	113.5	24.1
int NYT + PTB	0.6	107.8	24.7
int NYT + PTB	0.8	107.4	24.8
int NYT + PTB	1.0	104.5	25.0
All-Specific CL NYT +PTB	0.2	113.3	24.2
All-Specific CL NYT +PTB	0.4	108.4	24.6
All-Specific CL NYT +PTB	0.6	105.5	24.8
All-Specific CL NYT +PTB	0.8	104.6	25.0
All-Specific CL NYT +PTB	1.0	103.5	25.0

ing data suggests that curriculum learning is better in taking advantage of taking limited training data than linear interpolation. By using 80% yet-unseen training data, the adapted model trained using curriculum learning achieves the same performance as the adapted model trained using interpolation methods. In addition, using curriculum learning, only one final model is returned rather than two models as the interpolation method is based on. In other words, using curriculum learning method to deal with limited-data adaptation, we always only need to focus on one

model. However, using interpolation method, we probably need to deal with the risk to handle more models when more unseen data available.

We provide a final confirmation of the relationship of the performance improvements achieved in Table 9 are indeed due to an adaptation process. Specifically, we devise an anti-curriculum, which reverses the order of the All-then-Specific curriculum. Instead of training RNNLMs were trained from large existing data to small yet-unseen data, we train from a small data set (PTB data) to a large data set (NYT data). The resulting models were also tested on the PTB test data. The model trained in this way had a perplexity of 502.1 and a WPA os 20.2. Such an anti-curriculum is clearly ill-advised, since it does not even achieve the performance achieved using the NYT training data alone (i.e., NYT base in Table 9). We note that a major issue is the vocabulary, which, for reference, is 5% of the vocabulary size as NYT.

The results confirm the potential of curriculum learning to improve performance for limited-domain data adaptation of RNNLMs. In practice, the update of RNNLMs can be accomplished by the proposed *All-Specific* curriculum learning method only on the yet-unseen training data rather than retrain the RNNLMs using the combination of the existing training data and the yet-unseen training data.

The previous Table 9 shows the positive side of using curriculum learning in language model updating. However, just like two sides of the same coin, the advantages of curriculum learning also bring risks. According to curriculum learning, RNNLMs learn more from more recently presented data. In other words, RNNLMs also gradually forget what they learned from previous training data. As shown in Table 10, using *All-Specific* curriculum learning method in language model updating, the updated models achieve degraded performance on the

existing test data. However, the risk can be reduced by setting small learning rate and using the existing validation data together with new validation data. Basically, we need to trade-off the influence from existing data and new data. In the experiment, we also tested *Data-Sort* curriculum learning in language model updating in which the new data and the existing data need to be combined together. In other words, *Data-Sort* curriculum learning is more time consuming than *All-Specific* method.

Table 10: Comparison of RNNLMs trained on existing and yet-unseen data using the *All-Specific* curriculum learning method (*All-Specific* CL NYT +PTB) and RNNLMs trained using *Data-Sort* CL NYT +PTB curriculum learning method. Results (WPA) are reported on *existing* test data (NYT test data). The percentage of yet-unseen data used in training is given in the column marked 'ratio'.

ratio	*All-Specific* CL NYT +PTB	*Data-Sort* CL NYT +PTB
0.0	24.3	24.3
0.2	24.0	24.3
0.4	23.8	24.1
0.6	23.7	24.0
0.8	23.6	24.1
1.0	23.5	24.1

6. Conclusions

In this paper, we investigated the use of curriculum learning for the adaptation of RNNLMs. We focus on two situations for language model adaptation, namely, within-domain adaptation and limited-data within-domain adaptation.

To address within-domain adaptation, we use a component model method. Each component model is a sub-domain-adapted RNNLM trained by curricula that

were scheduled from general patterns to specific patterns. For within-domain adaptation, three different experiments have been used to investigate three different situations, namely the oracle situation (sub-domain information is known during training and testing), the situation that sub-domain information is known only in training and the situation that the sub-domain information is unknown both in training and testing.

Three different curriculum learning methods were proposed and analyzed, namely starting from the same vocabulary (*Start-Vocab*), data sorting (*Data-Sort*) and all then specific training (*All-Specific*). We compared the sub-domain-adapted models that are trained by these methods with conventional RNNLMs using a heterogeneous spoken Dutch data set. The results under the oracle condition under which sub-domain information is known at test time show that sub-domain-adapted models that were trained using *Data-Sort* and *All-Specific* outperform the conventional RNNLM. Especially on the "News" sub-domain, the sub-domain-adapted models achieved an over 50% reduction in terms of perplexity and a more than 30% improvement in terms of word prediction accuracy. The results reveal that curriculum learning can be used to shape the final RNNLMs to emphasize the specific patterns in the sub-domains. An additional experiment demonstrated that the gains achieved by RNNLMs are general: if the underlying RNNLM is improved, the curriculum learning adaptation method will translate the underlying improvement into an overall improvement.

When the sub-domain information is not available in testing, the sub-domain-adapted models need to be combined to make final prediction. On the sentence level, two combination methods were used. Using hard decision method, for each sentence, one sub-domain-adapted model that gave the maximum probability was

selected. Using soft decision method, for each sentence, a heuristic dynamic linear interpolation was used to combine the different sub-domain-adapted models. Experimental results show that the proposed model using these two methods outperform conventional RNNLMs.

Under the situation that the sub-domain information is unknown both in training and testing, the proposed models were tested using WSJ data set in N-best rescoring. During the training, sub-domain information was obtained by using Latent Dirichlet Allocation in conjunction with k-means clustering. The experimental results show that RNNLM adaptation using *Data-Sort* curriculum learning can achieve a limited, but not entirely unpromising, reduction in terms of word error rate.

To sum up, the set of experiments on within-domain adaptation shows that curriculum learning method can be used to train sub-domain-adapted models that emphasize the patterns characterizing specific sub-domains. Sub-domain-adapted models trained using curriculum learning outperform conventional RNNLMs on the corresponding sub-domains. When sub-domain information is unknown during testing, the combinations of the sub-domain-adapted models using a soft combination or a hard combination outperforms conventional RNNLMs and sentence level mixture RNNLMs.

To address limited-data within-domain adaptation, we use curriculum learning as an implicit interpolation method to combine patterns characteristic of existing training data with patterns characteristic of yet-unseen data. The results from our experiment on limited-data domain adaptation reveal that curriculum learning methods are more effective than conventional interpolation methods. The experiment also shows that updating of the existing RNNLMs with curriculum learn-

ing requires training only on the yet-unseen data without retraining models from scratch by adding the yet-unseen data to the existing data.

The comparison of these three types of curriculum learning proposed in this paper reveals that good performance is achieved by the curricula that are designed to be appropriate for specific data and specific challenges. In this paper, results on the WSJ data set and on the CGN data set did not achieve comparable improvement. For this reason, our future work will focus on the relationship between the design of the curriculum and the characteristic of the target data.

In this paper, we have chosen RNNLMs to be representative of neural network language models, under the assumption that the application of curriculum learning to other neural network language models would yield similar behavior. Our future work will explore this assumption in greater detail, allowing further insight onto how the method put forth in this paper should best be operationalized.

References

Amaya, F. A., Benedí, J. M., Jul. 2001. Improvement of a whole sentence maximum entropy language model using grammatical features. In: Proceedings of 39th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 10–17.

Bellegarda, J. R., 2004. Statistical language model adaptation: review and perspectives. Speech Communication 42 (1), 93–108.

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., 2003. A neural probabilistic language model. Journal of Machine Learning Research 3, 1137–1155.

Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning. In: Proceedings of International Conference on Machine Learning. ACM, pp. 41–48.

Berger, A. L., Pietra, S. A. D., Pietra, V. J. D., 1996. A maximum entropy approach to natural language processing. Computational Linguistics 22, 39–71.

Blei, D. M., Ng, A. Y., Jordan, M. I., 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., Lai, J. C., 1992. Class-based n-gram models of natural language. Computational Linguistics 18, 467–479.

Chen, S. F., 2009. Shrinking exponential language models. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North

American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 468–476.

Elman, J. L., 1993. Learning and development in neural networks: the importance of starting small. Cognition 48 (1), 71 – 99.

Gildea, D., Hofmann, T., 1999. Topic-based language models using EM. In: Proceedings of EUROSPEECH. pp. 2167–2170.

Iyer, R., Ostendorf, M., 1999. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. IEEE Transactions on Speech and Audio Processing 7(1), 236–239.

Iyer, R., Ostendorf, M., Rohlicek, J. R., 1994. Language modeling with sentence-level mixtures. In: Proceedings of the workshop on Human Language Technology. pp. 82–87.

Jelinek, F., Mrialdo, B., Roukos, S., Strauss, M. J., 1991. A dynamic language model for speech recognition. In: Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, pp. 293–295.

Kuhn, R., de Mori, R., 1990. A cache-based natural language model for speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (6), 570–583.

Mikolov, T., 2012. Statistical language models based on neural networks. Ph.D. thesis, Brno University of Technology.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.

Mikolov, T., Deoras, A., Kombrink, S., Burget, L., ernock, J., 2011a. Empirical evaluation and combination of advanced language modeling techniques. In: Proceedings of Interspeech. pp. 605–608.

Mikolov, T., Deoras, A., Povey, D., Burget, L., Cernocký, J., 2011b. Strategies for training large scale neural network language models. In: IEEE Workshop on Automatic Speech Recognition and Understanding. pp. 196–201.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S., 2010. Recurrent neural network based language model. In: Proceedings of Interspeech. pp. 1045–1048.

Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S., 2011c. Extensions of recurrent neural network language model. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 5528 –5531.

Mikolov, T., Zweig, G., 2012. Context dependent recurrent neural network language model. In: IEEE Workshop on Spoken Language Technology. pp. 234–239.

Oostdijk, N., 1999. Building a corpus of spoken Dutch.
URL `http://lands.let.kun.nl/cgn/.`

Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J. P., Moortgat, M., Baayen, H., 2002. Experiences from the Spoken Dutch Corpus project. In: Araujo (eds), Proceedings of the Third International Conference on Language Resources and Evaluation. pp. 340–347.

Qiu, L., Xu, J., 2013. A chinese word clustering method using latent dirichlet allocation and k-means. In: International Conference on Advances in Computer Science and Engineering.

Rosenfeld, R., 1994. Adaptive statistical language modeling: A maximum entropy approach. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

Rosenfeld, R., 1996. A maximum entropy approach to adaptive statistical language modeling. Computer, Speech and Language 10(3), 187–228.

Rosenfeld, R., Chen, S. F., Zhu, X., 2001. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. Computer Speech and Language 15 (1), 55 – 73.

Shi, Y., Larson, M., Jonker, C. M., 2013. K-component recurrent neural network language models using curriculum learning. In: IEEE workshop on automatic speech recognition and understanding. pp. 1–6.

Shi, Y., Wiggers, P., Jonker, C. M., 2010. Language modelling with dynamic bayesian networks using conversation types and part of speech information. In: The 22nd Benelux Conference on Artificial Intelligence. pp. 154–161.

Shi, Y., Wiggers, P., Jonker, C. M., 2011a. Combining topic specific language models. In: Proceedings of the International Conference on Text, Speech and Dialogue. pp. 99–106.

Shi, Y., Wiggers, P., Jonker, C. M., 2011b. Socio-situational setting classification based on language use. In: IEEE workshop on automatic speech recognition and understanding. pp. 455 – 460.

Shi, Y., Wiggers, P., Jonker, C. M., 2012a. Adaptive language modeling with a set of domain dependent models. In: Proceedings of the International Conference on Text, Speech and Dialogue. pp. 472–479.

Shi, Y., Wiggers, P., Jonker, C. M., 2012b. Towards recurrent neural networks language models with linguistic and contextual features. In: Proceedings of Interspeech. pp. 1664–1667.

Tam, Y.-C., Schultz, T., 2005. Dynamic language model adaptation using variational bayes inference. In: Proceedings of Interspeech. pp. 5–8.

van den Bosch, A., 2006. Scalable classification-based word prediction and confusible correction. Traitement Automatique des Langues 46 (2), 39–63.

Wang, W., Harper, M. P., 2002. The superARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In: Proceedings of Conference of Empirical Methods in Natural Language Processing. pp. 238–247.

Wiggers, P., Rothkrantz, L. J. M., 2006. Dynamic bayesian networks for language modeling. In: Text and Speech and Dialogue. p. 555–562.