Submodular data selection in ASR language modeling

Ahmed Aly

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2016

Reading Committee:

Katrin Kirchhoff, Chair

Gina-Anne Levow

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

Submodular data selection in ASR language modeling

Ahmed Aly

Chair of the Supervisory Committee:
Research professor Katrin Kirchhoff
Electrical Engineering department

Given the vast amount of textual data that we have available today, it is very beneficial to have an efficient methodology to filter and select important and relevant chunks of this data to improve current natural language and speech processing systems. Although utilizing very large language models has been the industry norm in the current automatic speech recognition production systems, the focus is now shifting towards efficient ways to generate and utilize personalized and adapted language models as they have proven to improve the end user experience. Submodular methods have achieved great success in different domains; acoustic modeling, text summarization, and machine translation. They provide a natural way to select high-quality relevant data from an out-of-domain data source to be utilized in domain adaptation and personalization. In this work, we model the problem of language modeling data selection as submodular function optimization. Our results show that indeed by using the submodular data selection methods we were able to train better language models with less data. We were also able to reduce the end-to-end word error rate of the ASR system 7% by selecting data from a completely different domain.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ASR: automatic speech recognition, is the process of automatically transcribing audio into a sequence of words.

SMT: Statistical machine translation.

LANGUAGE MODELING: probability distribution over sequences of words, it is used to estimate the likelihood of a given word appearing in a certain context.

N-GRAMS: sequences of words of length N.

PERPLEXITY (PPL): is a measure of how good the language model is able to estimate given sequences of words. Lower perplexity means better ability to estimate the sequence.

WER: word error rate, which is a metric to evaluate an end-to-end speech recognition system. It is defined as the ratio between number of mistakes made by the ASR system and the total number of words.

TF-IDF: Term frequency inverse document frequency, which is a statistical metric that measures the importance of a certain term to a given document by computing the ratio between the frequency of this term in the document and its frequency in the whole corpus.

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to the linguistics department in the University of Washington, where he has had the opportunity to work with great professors, and to his supervisor Katrin for her valuable time and guidance throughout the thesis.

DEDICATION

to my dear wife, Nehad and my great mother, Nagwa

Chapter 1
INTRODUCTION

Automatic Speech recognition (ASR) systems have been a very important area of research, especially in recent years. It has become the future of user interfaces, recent studies have shown that smartphone speech recognition can write text messages three times faster than human typing. This is driving many technology companies to utilize ASR systems to make their products speech-enabled, they believe that ASR is the way to enable more natural and faster user interfaces.

In order for users to rely on ASR in doing certain tasks while using their devices, it has to be very accurate and efficient, which is what we are trying to address in this work; improve the end-to-end accuracy of the ASR system in a computationally efficient way.

Traditional ASR systems have three main components:

- **Acoustic model**: Which models the relationship between the speech input and the set of phonemes that makes up this speech.

- **Language model**: Which captures the different variations and regularities of the natural language spoken by the users. It provides a way to score any sequence of words that the speech system will encounter so that the system can select the sentences with the top scores.

- **Decoder**: Which is the module that utilizes the acoustic model and the language model to map the speech input to the most likely sentences spoken by the users.

In this work, we will focus on the problem of language modeling. Our ultimate goal is to build better language models for some domain/application which can be utilized in the

ASR systems efficiently. Our approach for doing this is by selecting the most important text segments from the vast amount of text content that is available to us, which is not related to our target domain, and utilize this data to build better language models that improve the end-to-end word error rate (WER) of the ASR system. We formalize the data selection procedure for language model training as a submodular function optimization problem. Using this formalization we were able to build better language models with less data. We show that our method can be used very efficiently to select data from a completely different domain that has the same statistical properties as a limited in-domain or user specific dataset, which improves the WER of the ASR system.

Example: Assuming that we are trying to build a speech recognition system for a personal assistant which saves reminders, sets alarms, and answers questions about the weather. Usually in these applications, the in-domain training data is limited, and not enough to capture all the regularities and the variations of the data that we are trying to model. Also, it is very expensive to manually harvest and label in-domain data. Assuming that we have a huge text corpus of news articles, our method provides a way to automatically enrich the in-domain training data by selecting the best text segments from the out-of-domain data (the news corpus in this example) that matches the in-domain data. For instance, assume there are news articles discussing the weather in the news corpus, then we can benefit from these articles in building better language models that makes the personal assistant better at answering questions about the weather. The same is applicable for the other applications.

Submodular function optimization has been applied successfully in the problem of data selection for statistical machine translation (SMT) in [9]. It was able to outperform several baselines and widely used methods, and it also has mathematical performance guarantees that these other methods don't have. In Chapter 2 we will discuss the background and the different methods that are being used in the literature, in Chapter 3, we will discuss the submodular function optimization and our methodology of applying it to model the data selection problem for building better language models. In Chapter 4 we will describe our methodology in the experiments, the different variations that we have tried, the data and

the tools that we have used in our experiments and finally show our results. In Chapter 5 we will discuss the results that we obtained and their significance along with final remarks and promising research directions.

Chapter 2
LITERATURE SURVEY

In this chapter, I will discuss the different approaches and techniques that are widely used in the literature to select textual data to improve the performance of various NLP related applications.

2.1 Cross-entropy method

This approach was presented in [15]. It selects text segments from an out-of-domain data source based on the ratio between the cross entropy according to an in-domain model and out-of-domain model picked randomly from this data source.

Let I be an in-domain data set and O the out-of-domain data set. Let LM_I be a language model trained on I and LM_O be a language model trained on a random sample of O with the same size of I. Let $H_I(s)$ be the cross-entropy of a sentence s drawn from O according to LM_I; similarly, let $H_O(s)$ be the per-word cross-entropy of s according to LM_O. Each sentence in O is scored according to the difference between $H_I(s)$ and $H_O(s)$ Then all the sentences whose score is less than a threshold T are selected.

Thus each sentence s in the out-of-domain data is scored as follows:

$$Score(s) = H_I(s) - H_O(s) \tag{2.1}$$

Sentences are then selected based on a score cutoff optimized on in-domain model data. The reasoning behind this method is that; assume the out-of-domain data set is large enough that it contains an in-domain subset O_I that is drawn from the same distribution as the in domain data I. This subset will be a good candidate to select from O since it is statistically similar to I. The probability of selecting a sentence s from O such that s belongs to the subset

O_I will be:

$$P(O_I|s,O) = \frac{P(s|O_I,O)P(O_I|O)}{P(s|O)} \tag{2.2}$$

Since $O_I \subset O$, then $P(s|O_I,O) = P(s|O_I)$, also given the assumption that O_I and I are statistically similar, then we can replace $P(s|O_I)$ with $P(s|I)$, and thus the equation will become:

$$P(O_I|s,O) = \frac{P(s|I)P(O_I|O)}{P(s|O)} \tag{2.3}$$

By estimating all the probabilities on the right-hand side of this equation for each sentence, we can easily get the set of sentences with the highest probability to be in O_I. $P(s|I)$ and $P(s|O)$ can be estimated by training a language model on I and a random sample of O. The only remaining quantity is $P(O_I|O)$ which is not important to the quality of the selected sentences and can be omitted, and only the ratio $\frac{P(s|I)}{P(s|O)}$ affects the quality of the selected sentences.

By transforming this ratio to the log domain we get the following equation: $log(P(s|I)) - log(P(s|O))$, which is equivalent to equation (2.1) since $H_I(s) - H_O(s)$ is just a length normalized version of $log(P(s|I)) - log(P(s|O))$.

Advantages:

- This method scores all the sentences which scales well with the number of instances to select.
- Simple to implement
- Based on a simple intuition

Disadvantages:

- Doesn't eliminate duplicates
- Has no mathematical guarantees

2.2 In-domain perplexity method

In [6], they used a method similar to the cross-entropy method, in which only the perplexity according to the in-domain language model was used to score text segments from an out-of-domain data source. The candidate text segments with perplexity less than some threshold are selected.

2.3 Unigram model estimation method

In [10], a related method was used, in which a unigram language model was estimated from the full out-of-domain corpus. Then each text segment was scored according to the change in the log likelihood of the in-domain data according to the unigram model after removing this segment. The greater the decrease in the log likelihood, the more relevant the removed text segment to the in-domain data.

2.4 Linguistically-augmented perplexity-based method

In [21], they augmented the regular perplexity based selection methods with word-level linguistic units (i.e. lemmas, named entity categories and part-of-speech tags). The cross-entropy method was used as a baseline. In order to incorporate the more granular linguistic units, the text was transformed as follows:

- For named entities: Any word that matches a certain named entity was replaced by its category (e.g. America: Country)
- For lemmas: All the words are transformed to lemmas

Then the results from the different selection methods were interpolated together using two approaches:

1. Naïve selection of the top sentences from each method
2. Linear interpolation of datasets selected by the different methods

They reported that these word-level linguistic units were very beneficial especially for languages with high type-token ration (e.g. Chinese) or rich morphology (e.g. Czech) and less beneficial for remaining languages (e.g. English and Spanish).

2.5 Hybrid data selection model

In [22], along the lines of the previous approach of combining different selection methods to get better results, a similar approach was utilized that combines cosine tf-idf, perplexity, and edit distance methods by linearly interpolating their results. They concluded that each of these approaches has its own advantages and combining them together outperforms using single selection methods.

2.6 Data selection using neural language models

In [3], the authors extended the cross-entropy method that was introduced in [15]. They built their approach upon the same intuition of the cross-entropy method that to select relevant data from an out-of-domain data source, it should be statistically similar to the in-domain data and dissimilar to the average out-of-domain data. The new contribution of [3] is that recurrent neural language models [13] were used instead of the classic n-gram language models that were used in [15]. They applied their technique in the SMT data selection as follows:

For each pair of sentences in the out-of-domain data (e, f), where e is an English sentence and f is a foreign sentence, their score is:

$$Score(e, f) = [IN_E(e) - O_E(e)] + [IN_F(f) - O_F(f)] \tag{2.4}$$

where $IN_E(e)$ is the length normalized cross entropy on the English in-domain LM, and $O_E(e)$ is the length normalized cross-entropy of the sentence e on the English out-of-domain LM. Similarly $IN_F(f)$ and $O_F(f)$ are the length normalized cross entropy of f on the in-domain and out-of-domain foreign LMs respectively.

Similar to the original cross-entropy method, all the sentence pairs are ranked according to the score above and those with a score greater than an empirical threshold are selected to train the SMT system.

The 4 LMs that are used to calculate the sentence pair score in equation (2.4) are recurrent neural language models [13]. The main drawbacks of the ngram language models are that:

- They are not good at handling unknown ngrams. When they encounter an unseen ngram, they backoff to a lower order ngram which is very frequent in the adaptation data.

- They take a limited context into consideration when predicting the probability of the next word.

The recurrent language model was used to overcome these two drawbacks of the ngram language models;

- It does this by representing words using continuous vectors of fixed sizes (word embeddings) instead of the sparse word identity representation.

- Along with learning the probability of the words with each step, it learns a summarized continuous representation for all the previous words seen (hidden state).

This continuous representation of the words and the hidden state is more robust to rare contexts as it allows the sharing of parameters between similar words and contexts.

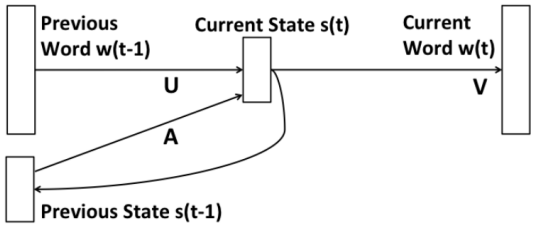

Figure 2.1: The RNN LM architecture used in [3]

To better model unkown contexts, they preprocessed the training data and replaced all the low-frequency words (with count < 2 in their experiments) by a special unkown token. This way there model was able to learn a proper embedding for the unkown token that can be used in run-time as the embedding for all the out-of-vocabulary words.

By utilizing the RNN language models, they reported end-to-end translation improvements from 0.1 to 1.7 BLEU compared to conventional n-grams.

Chapter 3

SUBMODULAR FUNCTIONS

3.1 Definition

Submodular functions are a class of discrete set functions that have the property of diminishing returns. In other words, they are functions for which their output (return) diminishes for the same input whenever the context in which this input is analyzed expands. Assume that we have a finite set of n objects V, and a valuation function $f(x) : 2^V \rightarrow R_+$ that evaluates any subset of V, and returns a positive real value for any subset $X \subset V$. We define $f(x)$ as a submodular function if it satisfies equation 3.1 below:

$$f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y) \tag{3.1}$$

Where $X \subseteq Y \subseteq V$. Assuming that the gain from adding a certain sample to the set X is g_1, and the gain from adding the same sample to the set Y is g_2, then the valuation function f is submodular if $g_1 \geq g_2$. Submodular functions have been applied successfully recently in different machine learning applications (e.g. in [11] and [8]), also they have been applied in natural language processing (e.g. document summarization in [12]) and speech applications (e.g. speech data selection in [23]).

3.2 Submodularity in SMT

The work that is presented in this thesis is inspired by the results of applying the submodular function optimization method in selecting training data for SMT systems presented in [9].

Submodular optimization was used for two main objectives:

- A natural way to summarize the training data and eliminate unnecessary redundancy

- It selects segments from an out-of-domain data which matches the in-domain training data the most

Since an exact solution for the submodular function optimization problem is NP-complete [4], a greedy approach [16] was utilized to get an approximate solution as shown in Algorithm 1:

Algorithm 1: Submodular functions greedy optimization

Input : Submodular function f : $2^V \rightarrow \mathbb{R}_+$, cost vector m, budget b, finite set V.

Output: X_k where k is the number of iterations

Set $X_0 \leftarrow \emptyset; i \leftarrow 0;$

while $m(X_i) < b$ **do**

  Choose v_i such that: $v_i \in \left\{ argmax_{v \in V/\ X_i} \frac{f(\{v|X_i\})}{m(v)} \right\}$

  $X_{i+1} \leftarrow X_i \cup v_i; i \leftarrow i + 1;$

end

Results in [9] have shown that utilizing submodular functions in SMT outperformed the widely used cross-entropy method while providing mathematical performance guarantees.

3.3 Submodularity in Language modeling

A natural extension to the current applications of submodular functions optimization is to apply it in selecting language modeling data for ASR systems. We found that the submodular functions and text features that were used in selecting the SMT training data can be easily extended and utilized in the ASR language modeling data selection. In this work, submodular function optimization was applied to select text segments from an out-of-domain data source to match an in-domain limited dataset in order to optimize the WER of a speech system. We formulate the problem as follows; we have an in-domain dataset X that contain text segments (sentences in our experiments) $\{x_1, x_2, x_3, ...\}$ and an out-of-domain dataset Y that contains

text segments $\{y_1, y_2, y_3, ...\}$ The ultimate goal is to select the best subset $Z \subseteq Y$ such that Z should satisfy the two conditions:

- $|Z| \leq B$ where B is some budget that we define (maximum number of words in our experiments)

- $Similarity(X, Z) > Similarity(X, V)$ Where $V \subset Y$, $|Z| \geq |V|$ and $V \neq Z$. Which means that the selected subset Z should be matching the in-domain set X more than any other subset in Y that has size $\leq B$

3.3.1 Submodular functions class

We have used the feature-based submodular function [19] that was utilized in [9] which has the form:

$$f(X) = \sum_{u \in U} w_u \phi_u(m_u(X)) \tag{3.2}$$

where $w_u > 0$ is a feature weight, $m_u(X) = \sum_{x \in X} m_u(x)$ is a non-negative modular function specific to feature u, $m_u(x)$ is a relevance score which indicates the relevance of feature u in object x, and ϕ_u is a u-specific non-negative non-decreasing concave function. Then the gain of selecting a new example v given that we have already selected a set X is $\sum_{u \in U} \phi(m_u(X \cup v)) - \phi(m_u(X))$

To apply equation 3.2 in the language modeling data selection problem, we defined the equation parameters as follows:

- **Features (U):** We have experimented with two settings for features: the first is to use the set of ngrams of the transcriptions of the ASR training set. The objective of this setting is to select text segments from the out-of-domain data that matches the ASR training set. The other setting is to use the ngrams of the output of the ASR system for the test set (see next chapter for more details)

- **Relevance scores ($m_u(x)$):** For each ngram u in a sentence x, we use its tf-idf which is computed as:

$$tf(x,u) = \sum_{i=1}^{|U(x)|} u_i = u$$

$$idf(X,u) = log(\frac{|X|}{\sum_{j=1}^{|X|} tf(x_i,u)})$$

$$\textit{tf-idf}(x,u) = tf(x,u) * idf(X,u)$$

  Thus the tf (term-frequency) part of this equation measures the frequency of the ngram u in sentence x, and the idf (inverse document frequency) part of the equation measures the log ratio between the total number of sentences and the total number of occurrences of the ngram u in the dataset to select from. The product of the two quantities is used as the relevance score for ngram u in sentence x.

- **Feature weights (w_u):** Following the same intuition of the cross-entropy method which is to select data that is not frequent in an average sample of the out-of-domain data set, we define the weights for each ngram to be the ratio between the ngram's count in the in-domain data set and its count in the out-of-domain set. Also we use another factor which has the form $\beta^{|u|}$ where $|u|$ is the length of the ngram u, this way we can favor longer ngrams since they are attributed to better WER since they can capture longer context.

- **Concave function (ϕ):** This is a non-negative, non-decreasing function that is applied per feature. The choice of the concave function is very important in this problem because:

  - The quality of the greedy approximation of the optimum solution depends on the curvature of the concave function. In [2], they have shown that when the submodular function has a curvature $0 < c < 1$, the worst-case guarantee of the greedy solution $= \frac{1}{c}(1 - e^{-c})$

– Also the curvature of ϕ controls the rate of information decay, thus the level of feature redundancy that we allow in our selected data. For instance, with more aggressive decay, the gain of selecting a sentence that contains ngrams that were seen before will vanish very quickly.

We have used the same concave function that was used in [9] which is the square root. Also experimented with other variations like $\phi(a) = a^{0.7}$ and $\phi(a) = a^{0.3}$ but we got better perplexities on a heldout set with the simple square root function.

3.3.2 Selection procedure

Since the out-of-domain corpus is very large, we have utilized the approximation of the greedy algorithm that was proposed in [14] which is a parallel distributed approach that doesn't require centralized evaluation of the full corpus for the selection.

Algorithm 2: Two-step parallel greedy procedure

1 Partition data into n partitions
2 Run the greedy algorithm on each partition in parallel
3 Merge the results from each partition and run the greedy algorithm on the merged set until the selected data size is equal to the budget

Chapter 4

SUBMODULAR DATA SELECTION EVALUATION

In this chapter, we will describe our work for evaluating the submodular data selection method in the task of ASR language modeling. We have used Google's 1-billion-word corpus [1] as the background corpus to select from, and the Switchboard data [7] as the in-domain data. We have used SRILM [20] to train our ngram language models, also we have used Switchboard recipes for Kaldi [17] to evaluate our generated language models in first pass decoding and lattice rescoring in different settings[1].

4.1 Baseline

In order to better understand the effectiveness of our approach and how is it compared to the other data selection methods; we have used the cross-entropy (Xent) method as a baseline for our experiments. The reasons for picking the cross-entropy method specifically are:

- It is widely used in many natural language processing applications
- Most of the other perplexity based methods are following the same intuitions that the cross-entropy method are based upon

For each experiment, we run both the submodular selection and the cross-entropy selection procedures, then we train a language model after adding the selected data to the training data. Our evaluation metrics are the perplexity of the trained language models and the end-to-end WER of the ASR system on the test data.

[1]Many thanks to Yuzong Liu who provided all Kaldi recipes and trained the acoustic models that we used in our experiments during his Ph.D in the University of Washington.

4.2 Data

4.2.1 Switchboard dataset

We have used the Switchboard-1 [7] (LDC97S62) dataset. It is a telephone conversation corpus that was originally collected by Texas Instruments in January 1990. It was first published by the National Institute of Standards and Technology (NIST) and distributed by the Linguistic Data Consortium (LDC) in 1992-3 in an effort to encourage speech recognition research by providing benchmark datasets for speech recognition researchers. The data itself is a collection of about 2,400 telephone conversations where two persons are involved in each conversation. 543 speakers (302 male, 241 female) from all areas of the United States were involved in order to capture as much accent variations as possible. A robot operator was responsible for handling the calls; 1- It gives the caller appropriate recorded prompts 2- It randomly selects and dials another person to take part in a conversation and it makes sure that the caller hasn't conversed with the callee before 3- It introduces a random conversation topic out of 70 predefined topics making sure that the two sides haven't spoken about the same topic before 4- It records the conversation on two separate channels until it is done.

For our language modeling experiments, we have split the transcription of these conversations into training data, heldout data and testing data. We have experimented with different data settings that we discuss in details in section 4.6

4.2.2 Google's 1-billion-word dataset

We have used Google's 1-billion-word dataset [1] as the out-of-domain data to select from. This dataset was introduced by Google to be a benchmark corpus for statistical language modeling research. The data was produced from the Machine translation workshop 2011 (WMT11) news crawl data, so it is completely different in nature from the Switchboard conversation data. Some statistics about the data:

Number of words	0.83B
Vocab size	793K

Table 4.1: Data stats

4.3 ASR system

4.3.1 Toolkit

We have used Kaldi toolkit [17] in our experiments; it is an open source speech decoder that maps the search space into WFSTs on different levels of granularity (e.g. phonemes, words).

4.3.2 Acoustic model

The acoustic model is DNN-HMM [18] system. There're 4 hidden layers in the network, with 1200 nodes in each hidden layer. The nonlinear function in the hidden layer unit is the tanh function. The output layer uses softmax function. The input layer takes a 40-dimensional fMLLR feature, followed by an additional splicing (+/- 4 frames), a de-correlation using LDA transformation (without dimensionality reduction). The fMLLR features are extracted using a SAT-GMM-HMM system [5].

4.4 Language modeling toolkit

SRILM toolkit was utilized to train the language models, we have tried both the Witten-Bell discounting method and Kneser-Ney discounting method, but the results reported here are using Witten-Bell discounting as it was yielding better results across all the models on the heldout data.

4.5 Data selection software

4.5.1 Submodular method software

The software that was used in the experiments in [9] was utilized. It is an implementation of the selection procedure that was described in section 3.3.2. We have parallelized the selection procedure over 40 threads which resulted in significantly reducing the selection time.

4.5.2 Cross-entropy method software

The cross-entropy method selection algorithm was written to produce and select the language models that we utilized in our experiments in this work. The high-level selection algorithm to select a set of size K words was as follows:

1. Select a random set of sentences from the out-of-domain data that matches the size of the in-domain data set

2. Build the in-domain language model using all the in-domain data

3. Build the out-of-domain language model using the random set selected in 1

4. For each sentence in the out-of-domain data calculate the ratio PPL(I) /PPL(O) where PPL(I) is the in-domain model perplexity and PPL(O) is the out-of-domain random model perplexity. Since the out-of-domain dataset is huge, this step was parallelized over 40 concurrent threads.

5. Sort all the sentences according to the ratio that was calculated in 4

6. Select the top N sentences until the requested size K is reached

4.6 Experiments

Our baseline is a language model trained on Switchboard training data. We then have two settings for data selection;

- Using Switchboard training data as the in-domain data in the data selection experiments. The resulting language models were evaluated in the ASR 1st pass decoding of the Switchboard test set.

- To evaluate the data selection methods in the adaptation scenarios, we have used the ASR hypotheses of the test set that were generated from the baseline experiment as the in-domain data. The resulting language models were evaluated in the lattice rescoring of the Switchboard test set.

We have limited the vocabulary in all of our language models to include only the vocabulary of the ASR system.

4.6.1 Model assembly

We have evaluated different models with different sizes; all the models include the baseline training data and then each of them contains the top X words in the selected data, where X is the batch size. We have tested with batches of sizes: 0.5M, 1M, 2M, 5M and 10M words.

4.6.2 Results

- Baseline (Using Switchboard training data to train an LM without any selected data)

Baseline-WER	Baseline-PPL
20.2	94.8

Table 4.2: Baseline results

- Using swb train as the in-domain data and using the selected data in the ASR 1st pass:

Size	WER(Submodular)	PPL(Submodular)	WER(Xent)	PPL(Xent)
0.5M Words	20.0	91.1257	20.2	94.6064
1M Words	20.0	89.4628	20.1	94.6789
2M Words	19.7	87.6397	19.8	93.7107
5M Words	19.5	85.7562	19.7	90.4786
10M Words	19.1	85.4337	19.1	85.2032

Table 4.3: Switchboard train as in-domain data

- Using ASR 1-best hypotheses as the in-domain data and using the selected data in lattice rescoring:

Size	WER(Submodular)	PPL(Submodular)	WER(Xent)	PPL(Xent)
0.5M Words	19.7	81.7236	19.8	90.1044
1M Words	19.5	76.6488	19.6	87.3587
2M Words	19.3	71.3688	19.5	84.0792
5M Words	19.0	66.4815	19.1	79.0287
10M Words	18.8	64.7025	18.7	76.6897

Table 4.4: ASR 1-best hypotheses as in-domain data

The first observation about the results is that we indeed see improvements in both perplexity and WER of the ASR system on the test data across the different selection configurations. Also we observed that in the second configuration; despite using the selected

data in lattice rescoring and not in the first pass, it is yielding better WERs than the first configuration. The reason is that there is a significant percentage of the test set ngrams not covered in the Switchboard training data (See next Chapter for more details). We tried to overcome this in the second configuration by using the 1-best hypotheses of the ASR system as the in-domain set, which indeed resulted in better WER and perplexities. Another interesting observation across all the experiments is that the improvements in the WER is not as significant as the improvements in perplexities, and in one case, the perplexity was improved but WER degraded by 0.1%. The reason for that is that perplexity is only impacted by the quality of the language model while the WER of the ASR system depends on many other components in addition to the language model and how they interact with each other; acoustic model, search vocabularies, lexicon and pruning parameters.

To better understand the effectiveness of the submodular selection method, we computed the number of unique ngrams that were selected in each of the experiments in Table 4.3:

Size	Number of unique ngrams(Submodular)	Number of unique ngrams(Xent)
0.5M Words	1393298	930546
1M Words	2628619	1859850
2M Words	4932714	3699549
5M Words	11262815	9076960
10M Words	20935317	17879340

Table 4.5: Number of unique ngrams in the selected data

We can see in Table 4.5 that the cross-entropy method tends to select more redundant ngrams (less unique) than the submodular method which explains the results that we are seeing in Table 4.3 and Table 4.4; thanks to the diminishing-returns nature of the concave function that we are using in the submodular method, it is able to select more relevant ngrams with less budget quickly during the selection, while the cross-entropy method stuck in

selecting redundant ngrams before it can expand to other relevant ngrams. We have observed that the effect of this phenomenon on the quality of the selected data diminishes as the size of the selected data increase beyond some threshold ($>$ 5M words in our experiments) since the cross-entropy method catches up on selecting the important ngrams when we increase the budget.

Chapter 5

FINAL REMARKS

In this chapter, I will discuss the main findings that I observed in the experiments, showing the significance of the results that we got and the scenarios in which our approach would be very beneficial. Also, I will discuss some promising extensions and next steps to the work presented in this thesis.

5.1 Conclusion

5.1.1 Data selection in ASR language modeling

Using data selection to build better language models from out-of-domain data corpus has proven to be very promising. Our results show that by efficiently selecting data from a huge news corpus we were able to achieve 7.4% relative reduction in the WER of the end-to-end speech system trained originally on human conversations which is a big improvement. This is very useful when we have a limited storage or run-time budget like the on-device environments; we can utilize the selection procedures that we used here to select the best data for on-device personalization and adaptation scenarios.

5.1.2 Submodular Vs Cross-entropy selection

We have noticed that for smaller selected data sizes ($<$ 5M words), the submodular method always outperforms the cross-entropy method. The reason is that the cross-entropy method tends to select text segments that contain duplicate ngrams that are very frequent in the in-domain data. With larger data sizes, the out-of-domain data runs out of this kind of ngrams, and the two methods tend to give the same performance.

5.1.3 Ngram coverage effect

We have also observed that the performance of the two methods is sensitive to the percentage of the ngram coverage of the evaluation dataset in the in-domain training data. To understand more why we don't see major improvements of the word error rate and the perplexity results when we use the Switchboard training data as the in-domain dataset, we have collected the following ngram coverage statistics in Table 5.1:

Ngrams	Total count	Not covered count	Out of coverage %
1-gram	3601	629	17%
2-grams	19566	6160	31%
3-grams	32506	17976	55%
4-grams	35623	28475	80%
5-grams	34129	32008	93%

Table 5.1: Ngram coverage of SWB Eval data in SWB training data

The big mismatches in the ngrams between the Switchboard test set and the training set reduces the effectiveness of the submodular and the cross-entropy selection methods when we use the training set as the in-domain data and evaluate the selected data on the test set (Table 4.3). The main intuition that the two methods depend on is selecting data that matches the in-domain data, in this case, there is a big mismatch between the in-domain data and the evaluation data that we are trying to evaluate the selected data on.

5.2 Next steps

A natural extension to this work is to augment the submodular method features with more granular linguistics features like the ones that were described in [21]. The easiest way to do that is to extend the set of features (U) that we use in equation 3.2 to include the lemmas, named entities and part of speech tags. By doing so, we can select not only text segments

that have surface similarities with the in-domain data, but also that share the structure and some level of semantics with the in-domain data.

Another area that we think will benefit from the submodular data selection is neural network language models. The reason is that training neural language models is very computationally expensive. Also, the computations increase exponentially with the size of the training data, which makes any reduction in the data size without affecting the quality of the language models a huge gain.

BIBLIOGRAPHY

[1] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2635–2639, 2014.

[2] Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.

[3] Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. Adaptation data selection using neural language models: Experiments in machine translation. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 678–683, 2013.

[4] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[5] Mark JF Gales. Semi-tied covariance matrices for hidden markov models. *IEEE Transactions on speech and audio processing*, 7(3):272–281, 1999.

[6] Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. Toward a unified approach to statistical language modeling for Chinese. *ACM Transactions on Asian Language Information Processing*, 1(1):3–33, 2002.

[7] John Godfrey and Edward Hollima. Switchboard-1 Release 2 LDC97S62. Web Download. *Philadelphia: Linguistic Data Consortium*, 1993.

[8] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1897–1904, 2011.

[9] Katrin Kirchhoff and Jeff Bilmes. Submodularity for Data Selection in Statistical Machine Translation. *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, 2014.

[10] D Klakow. Selecting articles from the language model training corpus. *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, 3:1695–1698, 2000.

[11] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4):1–20, 2011.

[12] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. *HLT '10 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (June):912–920, 2010.

[13] Toma Mikolov, Stefan Kombrink, Luka Burget, Jan Cernocky, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 5528–5531, 2011.

[14] Baharan Mirzasoleiman, Rik Sarkar, and Andreas Krause. Distributed Submodular Maximization : Identifying Representative Elements in Massive Data. *Advances in Neural Information Processing Systems*, 26:2049—-2057, 2013.

[15] Robert C Moore and William Lewis. Intelligent Selection of Language Model Training Data. *Proceedings of ACL*, (July):220–224, 2010.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.

[17] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011.

[18] Shakti P Rath, Daniel Povey, Karel Veselỳ, and Jan Cernockỳ. Improved feature processing for deep neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 109–113, 2013.

[19] Peter Stobbe and Andreas Krause. Efficient Minimization of Decomposable Submodular Functions. *Advances in Neural Information Processing Systems 23*, pages 2208–2216, 2010.

[20] Andreas Stolcke. Srilm an Extensible Language Modeling Toolkit. In *Proceedings of the Annual Conference of the Interna-tional Speech Communication Association, INTERSPEECH*, pages 901–904, 2002.

[21] Antonio Toral, Pavel Pecina, Longyue Wang, and Josef Van Genabith. Linguistically-augmented perplexity-based data selection for language models. *Computer Speech and Language*, 32(1):11–26, 2015.

[22] Longyue Wang, Derek Wong, Lidia Chao, Yi Lu, and Junwen Xing. ICPE: a hybrid data selection model for SMT domain adaptation. *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer Berlin Heidelberg*, 8202:280–290, 2013.

[23] Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. Submodular subset selection for large-scale speech training data. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 3311–3315, 2014.