IDIAP RESEARCH REPORT

LANGUAGE MODEL DOMAIN ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION

Amrutha Prasad Petr Motlicek
Alexandre Nanchen

Idiap-RR-05-2020

JANUARY 2020

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Language model domain adaptation for automatic speech recognition

Amrutha Prasad, Petr Motlicek, Alexander Nanchen
Idiap Research Institute

Report on internship of Amrutha Prasad at Idiap, May 2017
Report update, November 2018

Abstract

This report provides an overview of the work carried out in improving Language Model (LM) development used during the decoding of an Automatic Speech Recognition (ASR) system. The goal of this work is to develop a robust language model that can be adapted to multiple domains (ex: talks), offering better accuracies of the ASR system when applied to an adapted domain. By exploring and exploiting various datasets like Common Crawl, Europarl, news and TEDLIUM and by experimenting different techniques in training a model, we achieve the goal of adapting a general purpose LM to a domain like talks. This also significantly improves the ASR performance compared to the existing (generic version) LM.

1. Introduction

A statistical language model is represented by a probability distribution over sequences of words and is an important component of the Automatic Speech Recognition (ASR) decoding process. ASR system converts a speech signal to text which requires acoustic modeling and language modeling. During the decoding of speech signal, the acoustic model probabilities and Language Model (LM) probabilities are combined to produce a lattice (representation of alternative words sequences). During this process, the LM helps in providing the most accurate path and constrains the search among alternative word hypotheses. Hence, the quality of a LM is critical in an ASR system.

It is often impractical to build a new ASR system when data from unseen domain is processed. In such a case, adapting the existing LM to the new domain is feasible and is shown to provide higher recognition accuracy given the new domain[1]. A common strategy to build such a LM is to mine the text from web and select appropriately the text for the domain to which we wish to adapt to. The Common Crawl[2] dataset is a rich archive of web pages obtained by systematic web crawling freely available to the public. Such an archive can be expected to have a wide coverage of domains that can be used either to build a new LM or adapt an existing LM.

Data preparation is an important step when using textual corpora such as Common Crawl. We first perform text normalization which consists of removing the punctuations, converting the numbers to written numbers (ex: 43 to forty-three), etc. Once the data is normalized, we a known vocabulary is exploited to perform data (sentence) selection and train an n-gram statistical (i.e. ARPA format) model used in the ASR decoder. In this report, TEDLIUM test dataset is employed. Different data selection techniques are exploited to build statistical LMs for ASR decoder..

The report is organized into the following sections: 2. Data collection, 3. Data normalization, 4. Language Modelling, 5. ASR decoding, 6. Discussion.

[1] Bellegarda, Jerome R. "Statistical language model adaptation: review and perspectives." Speech communication 42.1 (2004): 93-108.

[2] http://commoncrawl.org/

2. Data Collection

2.1 Out-of-domain (background) data

Training a new LM requires a large corpus of text data. Such large datasets, also called background data, usually cover multiple (linguistic) domains to keep the model as generic as possible, or domain-independent. Building domain-independent LM is usually followed by a process of adaptation, retraining the generic LM to a target domain. The process of adaptation requires additional text data collected from in-domain resources.

Web crawls are often good sources for background dataset. Common Crawl (CC)[2] is a non-profit organization which provides copy of the entire web to the public. Typically this includes Petabytes of data. In our following experiments we have used a subset of CC[3] to train a LM. Initially we downloaded a file which was 50GB[3] and split it into 10 files of 5GB each.

2.1 In-domain data

To adapt a LM to a target domain (e.g. TED talks considered as target data in this work), additional textual data is required. More specifically, this work uses three different corpora: TEDLIUM transcripts, Europarl transcripts, and news transcripts. TEDLIUM[4] is a series of TED talks provided by the Laboratoire d’Informatique de l'Université du Maine (LIUM). This is partitioned to train, dev and test sets. Goal of this work is to adapt the LM developed using large CC resources, further adapted to target-domain so that the final ASR can provide superior performance on TEDLIUM test data. As the TEDLIUM train set is insufficient for adaptation, we also include the Europarl[5] and news commentary data[6].

3. Data Normalization

The first step in building a LM is to normalize the data. The process of normalization is done on all textual data i.e. the background data and the domain data. Normalization of the data is performed to:

- avoid the use of mixed-case text, specifically different versions of the same word. Otherwise, "Hello" and "hello" will be treated as different words.
- Remove punctuations, since it can introduce irrelevant word variants. Otherwise "therefore," and "therefore" will be treated as different words.

[3] This report focuses on English thus only English-related data resources are considered here.

[4] http://www.openslr.org/7/

[5] http://www.statmt.org/europarl/

[6] http://data.statmt.org/wmt17/translation-task/training-parallel-nc-v12.tgz

    - However We retain certain punctuations if they are in the vocabulary. For example: they’re is kept as another pronounciation variant in the vocabulary, in addition to ‘they’ ‘are’.
    - We replace certain symbols by words. Ex: % → percent.
- Convert all numbers to words. For example, "23" is converted to "twenty three".
- Acronyms are retained as it is with small letters.
    - Ex: A. I. R. is changed to a. i. r.

After the normalization process, the background data is reduced to approximately 44GB. The normalization is applied to the domain data as well. The ASRT[7] toolkit was used for this process. The toolkit supported French, German and Italian languages previously. The feature to support English language was added during this project.

Following table summarizes textual corpora used in this work. TODO

	Amount of data	
DataSet	**No. of sentences (K)**	**Quantity (in MB)**
Common Crawl	50’000	6’000
Europarl	N/A	250
News commentary	N/A	N/A
TEDLIUM	194	16.3

Table 1. Overview of textual resources exploited in this work.

[7] It is an open source and available at **https://github.com/idiap/asrt**

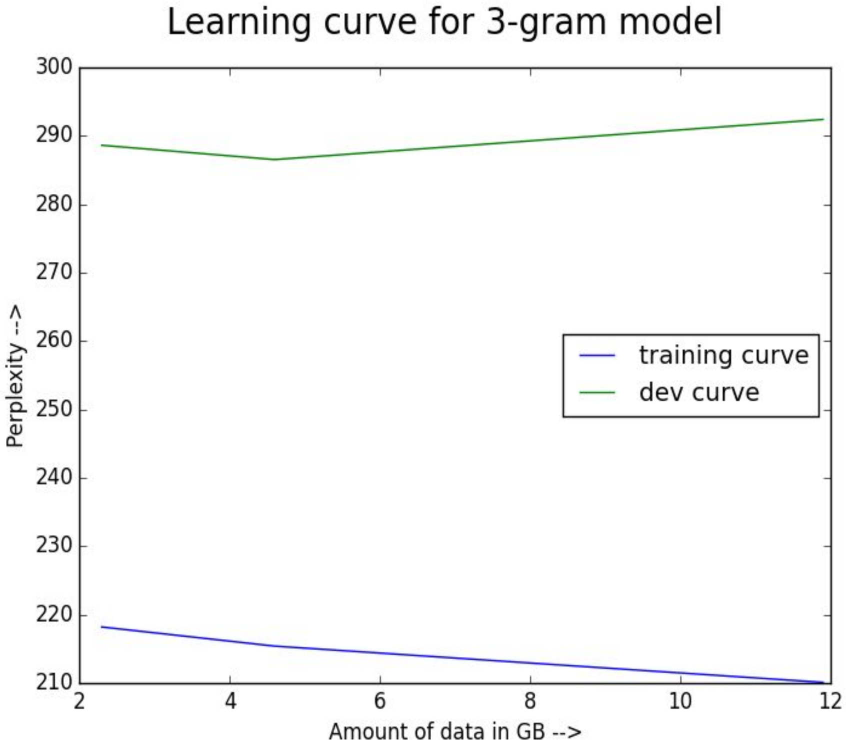

Figure 1: Perplexity graph for different amounts of data. The training curve represents the perplexity measured for the LM trained with different amounts of background data using the same data. The dev curve represents the perplexity measured for the LM trained with different amounts of background data using the TEDLIUM dev set.

4. Development of language Model

In this section we describe the process of sentence selection as well as the process of training several n-gram based LMs. In general, sentence selection helps to reduce the noise in the data, which results in smaller but cleaner textual corpora used to build the final LM. LM interpolation is one way of adapting domain-independent LM to domain-dependent LM.

Sentences for building the n-gram model are selected from the background dataset by using a known vocabulary[8]. A minimum percentage of coverage of the words is specified for a particular sentence to be selected. This is performed to ensure that the words in the selected text match (with large significance) the vocabulary. For example, if the coverage is set to 90% and a sentence has 10 words, 9 out of 10 words should be included in the pre-defined vocabulary. After this selection process, the background data is reduced from 44GB to roughly 20GB. The data is split into 10 files each of size 2GB. Each of these 10 files corresponds to the initial split mentioned in Section 2.1.

To further reduce the textual (background) data while exploiting data relevant to the domain, we performed the following experiments: we started using the first split (i.e. 2GB file) and gradually added other 2GB splits to increase the development set up to 12GB. Due to memory constraints, we

[8] /idiap/home/pmotlic/TEMP7b/UBM-ASR_12_2014/kaldi-trunk/egs/icsiami/s1/data/lang

did not increase the size further. At each stage, a 3-gram LM was developed and the perplexity was measured as shown in figure 1. We observed that after exploiting 6GB (3 splits) of data, the perplexity of the model saturated and no further significant improvements were obtained. For the final step to develop the LM, we have used 6GB of background data.

With the selected sentences performed in this Section 4, we can now train an n-gram LM. We started with a 3-gram model developed on the 6GB data using the Mitlm[9] toolkit. This model was further pruned using the Irstlm[10] toolkit by providing a threshold of "1.5e-8" to obtain a model of size of around 200MB (i.e. the original non-pruned model was of size about 8 GB). Process of pruning is usually required as the trained model could be of large size and thus could not be directly used to build a finite state transducer (decoding graph[11]) for the ASR decoding process. We specified a threshold to reach a usual model size (several hundreds of MB) that can be easily used to build the graph for decoding.

Two different LMs were developed separately. First, we trained a 3-gram model with background (Common Crawl) data, Europarl and news data. The second LM was trained with only TEDLIUM train data. Finally, both the LMs are interpolated[12] and pruned to the final LM size of around 200MB.

5. ASR decoding

Our goal is to explore the best way to adapt a generic LM (built using large amount of background CC data) to offer the best speech recognition performance on the TEDLIUM test set. We investigate different strategies to find the best LM that is generic enough but at the same time has a power to perform well on in-domain data. Word Error Rate (WER) measure is used as a final metric to evaluate performance of the adapted speech recognition system exploiting the developed LMs. In the following experiments the acoustic model is fixed. The acoustic model (conventional HMM/DNN hybrid built in Kaldi[13]) is trained on a subset of LIBRISPEECH, AMI and TEDLIUM datasets on 16kHz speech data. In total, around 150 hours of data is used for acoustic model training[14].

Below we describe different LMs being developed in the following experiments:

1. Baseline LM: the 3-gram LM that was used in the baseline ASR system[15]. It is trained with textual resources from LIBRISPEECH, ICSIAMI and TEDLIUM corpora. The 129K words vocabulary used does not comprise all the words (i.e. from test set), hence the TEDLIUM test set has an issue with Out-Of-Vocabulary (OOV) words.

[9] The github link to mitlm is **https://github.com/mitlm/mitlm**

[10] We apply standard IRSTLM toolkit to prune the initial LM   (https://sourceforge.net/projects/irstlm/).

[11] Kaldi constructing decoding graph **http://kaldi-asr.org/doc/graph.html**

[12] We apply SRILM to interpolate LMs.

[13] http://www.kaldi-asr.org

[14] **Investigating Cross-lingual Multi-level Adaptive Networks: The Importance of the Correlation of Source and Target Languages**, *Alexandros Lazaridis*, *Ivan Himawan*, *Petr Motlicek*, *Iosif Mporas* and *Philip N. Garner*, in: Proceedings of the International Workshop on Spoken Language Translation, Seattle, WA, USA, 2016.

[15] The ASR system that was used by Alexandros (/idiap/temp/alaza/ASR/librispeech/s5_v2/exp_with_tedwicsiami/tri4_sat_ivan_adapt_only_topbottom_layer_with_tr_train_cv_dev_tedlium_s5v3_2)

2. In-domain LM: the model is trained with in-domain TEDLIUM train data only. The data consists of 16.3MB, 193’787 sentences and 3’030’502 (total) words. The final 3-gram LM comprises approximately 40K words (unique words from the train and test TEDLIUM dataset). The test vocabulary was also included to avoid the OOV problem.
3. Out-of-domain LM: the model trained with CC, Europarl, news commentary, which approximates to 50 million sentences using the vocabulary of 129K unique words. Approximately there are 2K words from test set which are OOVs.
4. Out-of-domain LM with limited vocabulary: the model trained with Common Crawl, Europarl and news commentary which approximates to 50 million sentences with the TEDLIUM vocabulary of 40K.
    a. Here we do sentence selection process on the normalized CC data with the 40K vocabulary to find sentences related to the TEDLIUM domain.
5. Universal LM: the model trained with CC, Europarl, news and TEDLIUM train data which approximates to 52 million sentences with a vocabulary of approximately 140K (120K + 40K) unique words.
6. Interpolated LM: the models developed in (2) and (3) are interpolated and pruned (i.e. the final size is around 199MB).

The results of performance of the above experiments to decode the TEDLIUM test set are given in following Table 2.

Language Model	**WER(%)**	**Number of words with errors in test set** **Tot. words for estimating WER(N) = 27’512**
1. BASELINE	14.9	4’050
2. TEDLIUM	9.4	2’589
3. CC + Europarl + news (one model built)	16.5	4’554
4. CC + EU + news (41K) (one model built)	14.9	4’110
5. CC + EU + news + TEDLIUM (one model built)	15.5	4’265
6. CC + EU + news interpolated with TEDLIUM	9.7	2’658

Table 2: Different model performances on the TEDLIUM test set.

6. Discussion

With the baseline model described in Section 5, WER of the baseline ASR incorporating the LM built from several (TEDLIUM, ICSIAMI, LIBRISPEECH) textual corpora is around 14.9%. Using the TEDLIUM model, the WER reduced to 9.4% which gives a relative improvement of 37% compared to the baseline model. As expected, we see that the ASR performance degrades with an increase in

WER of 16.5% while using a model with no in-domain data for training. Using the same out-of-domain data with only the in-domain vocabulary performs similar to the baseline with a WER of 14.9%. Despite adding the in-domain data to the background data and training a single LM, the ASR performance is still considerably worse. This is because the in-domain data is significantly less compared to the background data and there is not much weightage given to the in-domain data. Therefore, using a generic model and interpolating with an in-domain model gives better ASR performance as it is possible to weight the influence of each of these models. The WER significantly reduced compared to the baseline and closer to the in-domain model.