Sādhanā (2018) 43:209
https://doi.org/10.1007/s12046-018-0976-x

Ameliorated language modelling for lecture speech recognition of Indian English

DISHA KAUR PHULL* and G BHARADWAJA KUMAR

VIT University, Vandalur–Kelambakkam Road, Chennai 600127, India
e-mail: dkphull@gmail.com; bharadwaja.kumar@vit.ac.in

MS received 21 October 2016; revised 17 March 2017; accepted 29 August 2018

Abstract. A great amount of research is growing towards the automatic transcription of lectures that consist of numerous information and knowledge that could be helpful to the educational systems and institutes. In large vocabulary speech recognition, language model plays a paramount role in reducing the humongous search space. However, language modelling is very brittle when moving from one domain to another or when moving from read speech to spontaneous speech. Also, lecture speech recognition will have some of the characteristics of spontaneous speech. Hence, it is very challenging to build the language model for this task. In this paper, a judicious approach to adapt the language model in a way where the language model will be in close proximity to the topic spoken in the lecture speech has been depicted. The evaluation of the language model is devised using the proposed approach with the existing language models such as CMU Sphinx, Gigaword and HUB-4. We observed the results analysis that the language models devised from the proposed approach outperform from the existing language models in terms of word error rate, perplexity and out of vocabulary rate. Analysis shows that the presented two-phase approach has resulted in an average decrease of the word error rate to be approximately 14% and the perplexity is decreased by half on average.

Keywords. Adapted language model; Indian English; information retrieval; lecture recognition; speech recognition.

1. Introduction

A lecture is an oral presentation intended to present information about a specific subject. Lectures are used to convey critical information, history, background, foundations, conditions and speculations. Nowadays, lectures related to most of the topics are available in University's website and it is expanding ceaselessly. Apart from the these websites, there are many platforms that provide lectures on an extensive variety of topics, which are available in many languages. NPTEL [1] provides e-learning through web, where the audio and video of the lectures given by the speakers with different varieties of Indian English (IE) are available. IE is the English spoken as the second language in India [2]. IE diverges incredibly as per the territorial and geological region in which the speaker resides. This is due to the fact that in India, there are 150 languages among which 23 are official languages, including English, and approximately 2000 dialects are spoken [3]. These subtle elements delineate the distinction in articulation of IE from different Englishes as it has the L1 (native language) influence.

In this paper, lecture speech recognition is performed on the lectures extracted from NPTEL. The lecture speech recognition task comprises many challenges such as the speaking style, the environment and the vocabulary [4]. In addition to the challenges, every lecture speech dwells on a blend of filler words (ok, like), filled pauses (um, ah), partial words, redundancies, disfluencies, word contraction (aren't), word reduction (wanna), co-articulation and assimilation. All these inflate the difficulties and hinder the efficiency of the lecture speech recognition task. Numerous efforts have been made recently for the enhancement of the lecture speech recognition systems. In particular, research contribution concentrating on the language modelling module for the lecture speech recognition task has been carried out ardently. Statistical *n*-gram language model (LM) [5] has been hugely exploited, which is found to be successful in the speech recognition task and for similar applications. Spontaneous and multi-domain speeches, such as lectures, encompass a constant change in language characteristics. Speech recognizer performance is severely affected when the linguistic characteristics of the discourse in the training and recognition tasks differ. This would require an LM that can adapt and update itself according to the speech domain, which is the motivation for our current work.

*For correspondence
Published online: 30 November 2018

In [6], the need for separate acoustic models and the importance of pronunciation dictionary for IE were investigated. The necessity for better language modelling in lecture recognition was emphasized and we have also contemplated that prior knowledge on topic can increase the performance of LVCSR. In general, the efficiency of the LMs depends largely on the availability and accessibility of the corpora specific to domain, topic, theme and style. However, in a generic recognition framework, prior knowledge about the topic spoken is unknown. Hence, to match the dynamic drift in the topic being spoken, an architecture for dynamic adaptation of LM has been proposed in this current work. In the present work, we specifically focus on devising a better language modelling technique for lecture speech recognition task. Here, a disparate framework is proposed to dynamically adapt the LM for increasing the lecture speech recognition's performance. The ennoblement of our LM is its ability to dynamically adapt in accordance to the topic being spoken, which has been capacitated due to the inclusion of a quintessential corpus such as Wikipedia dump and other web sources that are representative of a wide variety of domains.

The remainder of this article is organized as follows. Section 2 gives an account of related work regarding language modelling and lecture recognition. The overall methodology related to the two-phase recognition framework is described in section 3. Section 4 shows the experimental results for the phase 1 (P1), phase 2 (P2) without bootstrapping and with bootstrapping alongside comparative results for evaluating from the existing LMs. Finally, section 5 gives conclusions for language modelling in automatic lecture speech recognition system for IE.

2. Related works

From the literature survey, it has been observed that there is significant need for adapted language modelling while dealing with lecture recognition scenario. The language modelling in speech recognition followed different methods, procedures and frameworks, according to the problem in hand, to achieve better performance. Here, we discuss a few works related to the language modelling available in the literature.

There are many research works that explain the efforts on the information retrival for the topic-specific, adapted and dynamic LM incorporated in the Automatic Speech Recognition (ASR) system for better recognition. The work of Echeverry-Correa *et al* [7] comprises two tasks: topic identification and dynamic LM adaptation. The topic has been identified by the highest similarity of the words to the documents, for which they have used vector space model and latent semantic analysis, which have been predetermined [7]. Approximately 9% decrease in word error rate (WER) from the base has been shown. Watanabe *et al* [8] have proposed the topic tracking LM, which is applied for unlabelled incremental adaptation of LM. They have dealt with the topic changes caused by scene switching from topic to topic; they have also suggested that word probability θ plays an important role for latent topic to change gradually over time [8]. This shows a decrease of 1.5% in the WER on the MIT corpus and 1.3% decrease in WER on the corpus of spontaneous Japanese. Novotney *et al* [9] have explained the benefits of semi-supervised LMs for under-resourced languages and over a range of low resource condition. They have given limitation to back-off LMs and have motivated the robust use of automatic counts prior to the estimated parameters of a log–linear LM [9]. Oger and Linarès [10] have used the possibility theory for adapting the LM. They have used the data directly from the web for generating the LM [10]. The results are obtained by the 100-best-decoding process, which has provided an absolute WER reduction of 5.9% on AVISON and 1.1% absolute WER reduction on HUB4 task. Chen and Chen [11] have used information retrieval to generate the relevance LM. They have used it on the Chinese character recognition by finding the co-occurrence and the latent topic modelling, which creates the relevance models. This has helped in the decrease of Character Error Rate (CER) and the perplexity. In order to determine the ranking of documents according to the relevance, information retrieval from the queries has been performed. The web retrieval [12] of the documents according to the statistical approach was used for generative relevance language modelling. This approach has a noticeable performance improvement in all domains across all the evaluation metrics.

Latent Dirichlet Allocation (LDA) technique has been widely exploited to form topic-specific LMs. LDA with stem information (morphology) has been used for the inflectional languages, which showed better results, but has not showed much difference for English language [13]. Haidar and O'Shaughnessy have used LDA for unsupervised LM by adapting it from the test document [14]. They used the Matlab Topic Modeling Toolbox, which in turn resulted in a decreased WER due to the usage of the adapted LM [14].

Many linguistically rich languages have come up with the word-level linguistic approach for the generation of a better LM. Toral *et al* [15] have used word-level linguistic units such as lemmas, Named Entity Recognition (NER) and Parts of Speech (POS) tags. In this paper [15], two kinds of LMs are created: domain-specific corpus and random subset of general corpus, which is of same size as the domain-specific corpus. The perplexity reduction has been from 7% to 13% approximately. Karpov *et al* [16] have come up with the syntactico-statistical LM for large vocabulary Russian Speech Recognition task. In [16] they have searched for the syntactic word dependences in the sentence joining with the statistical LM to create bigram LM. There has been an absolute WER decrease of 3.4%, and 1% of absolute

decrease in the Letter Error Rate (LER). The phrase level paraphrase model has been statistically formed from the standard text with no semantics and generated into multiple paraphrase variants by maximizing marginal probabilities. A multilevel LM has been developed by incorporating both word level and phrase level information to form the paraphrastic LM [17], which has helped in a significant improvement in accuracy for ASR system.

Recently, neural networks (NNs) and deep networks were keenly applied in language modelling. Liu *et al* [18] have proposed a cascaded network based on NNLM (Neural Network Language Model) adaptation scheme. The cross-adaptation has been performed using a context-dependent multilevel LM at both syllable and word levels [18]. NN adaptation on the word level has been performed and then linear combination of the two LMs has been executed. This has helped in 4–7.1% decrease in the WER.

In lecture speech recognition, the web-based language modelling for topic-specific LMs has been used by making web as the source to filter out the data specific to a topic [19, 20]. These web-based topic-specific data have been found out to be very helpful in lecture transcription for adapting LM by sending proper query for search, which would return related documents and in turn rapidly generates the LM similar to the topic of the lecture. For improvements in lecture speech recognition, the information from the presentation slides is extracted [21]. The data extracted from the presentation slide thus are adapted with the baseline LM, which helps improve the overall accuracy by approximately 3%.

3. Methodology

This section describes the methodology and the steps involved in formalizing adapted LMs. In this section, we intend to focus on the framework that might ensure that the information related to the lecture is smeared into the LM. The statistical language modelling with Kneser–Ney smoothing has been considered to create the LM, which is discussed in section 3.1. The corpus utilized for creating the model requires a few preprocessing steps as elucidated in section 3.2. The P1 approach provides the baseline LM, which is used for different lecture topics and is explained in section 3.3. The P2 approach is described in section 3.4, through which the specification about the lecture topics could be noted. Each lecture topic consists of separate and distinct LM adapted specifically for it.

3.1 *LM*

The aim of this paper is to come up with an appropriate framework to enhance the productivity of the lecture speech recognition system by LM adaptation. LMs would help any speech recognizer to figure out how likely a word sequence is independent of the acoustics. The LM plays a vital role in resolving acoustic confusions that arise due to the occurrence of co-articulation, assimilation and homophones while decoding. Hence, LMs play a major role in guiding and constraining among large number of alternative word hypotheses in continuous speech recognition. In state-of-the-art speech recognizers, n-gram LM is still the predominant choice. The perception of the n-gram model is to compute the probability of a word by considering the history of last few words instead of its entire history. In an n-gram model, the probability $P(w_1, \ldots, w_m)$ of observing the sentence $w_1, \ldots, w_m$ is approximated as shown in Eq. (1):

$$\begin{aligned} P(w_1, \ldots, w_m) &= \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \\ &\approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}). \end{aligned} \tag{1}$$

The estimation of these n-gram probabilities is performed using the maximum likelihood estimation (MLE). The conditional probability can be calculated from n-gram model frequency counts (c) as shown in Eq. (2):

$$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{c(w_{i-(n-1)}, \ldots, w_{i-1}, w_i)}{c(w_{i-(n-1)}, \ldots, w_{i-1})}. \tag{2}$$

If any n-gram occurs a sufficient number of times, it would have good estimate of its probability but due to sparse data, we have the problem of zero-probability n-grams. The MLE method also produces poor estimates when the counts are non-zero but still small. Smoothing technique helps improve the poor estimation by discarding the zero probabilities. A common approach is to generate a maximum-likelihood model for the entire collection and linearly interpolate the collection model with a maximum-likelihood model for each document to create a smoothed LM. Kneser–Ney smoothing [22] helps in considering a lower order probability distribution that could be modified to account for that modelled by a higher order probability distribution. The Kneser–Ney smoothing uses the absolute-discounting interpolation, which assimilates the information of higher and lower order LM [23]. Absolute discounting is a much better method for computing the $c*$ based on the frequencies by subtracting a fixed discount δ from each count. A mix of probability estimates from all the n-gram estimators is performed during interpolation.

Let $c(w, w')$ be the number of occurrences of the word w followed by the word w' in the corpus; the uni-gram probability is shown in Eq. (3):

$$p_{KN}(w_i) = \frac{|\{w' : 0 < c(w', w_i)\}|}{|\{(w', w'') : 0 < c(w', w'')\}|}. \tag{3}$$

The uni-gram Kneser–Ney probability is the number of unique words the uni-gram follows divided by all bigrams,

and here λ is the normalizing constant. The bigram probability is shown in Eq. (4):

$$p_{KN}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - \delta, 0)}{\sum_{w'} c(w_{i-1}, w')} + \lambda_{w_{i-1}} p_{KN}(w_i). \tag{4}$$

Here, δ is a fixed number that is to be discounted over n-gram counts. These equations can be extended to n-grams as given in Eq. (5):

$$\begin{aligned} p_{KN}(w_i|w_{i-n+1}^{i-1}) = & \frac{\max(c(w_{i-n+1}^{i-1}, w_i) - \delta, 0)}{\sum_{w'} c(w_{i-n+1}^{i-1}, w')} \\ & + \frac{\delta}{\sum_{w_i} c(w_{i-n+1}^{i})} \\ & \times |\{w' : 0 < c(w_{i-n+1}^{i-1}, w')\}| p_{KN}(w_i|w_{i-n+2}^{i-1}). \end{aligned} \tag{5}$$

Kneser–Ney smoothing is a widely considered and most effectively used smoothing technique. For our work, we have considered this smoothing technique for creating the trigram LMs. Typically, n-gram models for large vocabulary speech recognizers are trained on hundreds of millions or billions of word strings. In constructing such kind of models, we usually face two problems. Firstly, the large amount of training data can lead to larger n-gram LM, which consequently leads to excessively large hypothesis search space. Secondly, to train a domain-specific model, we must deal with the data sparseness problem, because large amount of domain-specific data is not available.

In this paper, we have tried to come up with an adapted LM that would help better in performance. We have used the topic combination approach to dynamically adapt the LM required for any given lecture speech. The adaptation of the LM can be performed only after the first phase of recognition. For the present work, a constant interpolation value is considered after repeated trial and error method. The following sections briefly explain the process involved in creating the LMs.

3.2 *Preprocessing*

The documents obtained automatically are preprocessed so that the relevant and clean information could be obtained. The preprocessing helps in converting documents into the required manner for further processing. The following preprocessing steps were followed in both P1 and P2 of the recognition tasks for generating the LMs.

- **Abbreviation:** The known abbreviations are normalized to their expansions.
- **Sentence boundary:** The starting and ending of the sentence in the text are marked.
- **Alphanumeric:** All the alphanumeric words present in the documents are converted into non-alphanumeric forms.
- **Lexicalization:** All the numbers/digits are converted into the alphabetical representation.
- **Punctuation:** All the punctuation and extra white spaces present in the text are removed.
- **Case:** All the letters in the document are converted into lowercase as it has been assumed that uppercase and lowercase have no difference while considering LM for speech recognition.

3.3 *P1*

The LM for P1 has been formed in such a way that it should be generic and can also be used for all the topics/domains for the recognition task. The preprocessing of the base corpus and the Wikipedia dump is carried out by following the steps discussed in section 3.2. The P1 process is presented in figure 1. Generally, it is conceived that the higher the order of n-grams, the better the LM [19, 20]. With this conception, we have built trigram LM on the transcribed base corpus, which has spoken language characteristics. Later, we built trigram LM from Wikipedia corpus. However, the Wikipedia consists of nearly 1M words; hence, to create a vocabulary for LM, restriction on the words should be maintained [24]. Since Wikipedia is a multifarious corpus, most frequent words of the corpus can be considered as generic or common. With this assumption, we built the trigram model with the words having frequency more than 100 (amounts to 64000 words), which resulted in nearly 50M trigram combinations. Distressingly, the trigram model gave an out of bound index during the decoding process. This restriction enforced us to use the bigram model from the 64000 words. Finally, we interpolated the trigram model from base corpus and bigram model from the Wikipedia corpus for use as a generic LM.

Let $P(w|h)$ be the probability, where w is the word and h is the history of the sequence of words. Here, $P_b(w|h)$ is the base LM and $P_{wd}(w|h)$ is the Wikipedia dump LM.

$$P_{bw}(w|h) = \lambda_b P_b(w|h) + \lambda_{wd} P_{wd}(w|h) \tag{6}$$

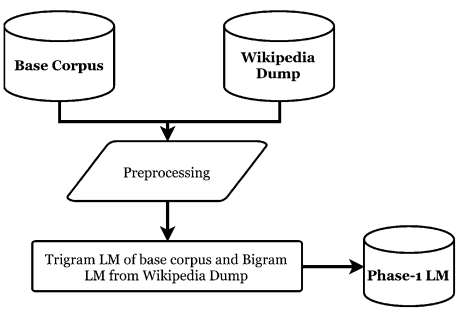

Figure 1. Phase-1 approach for LM generation.

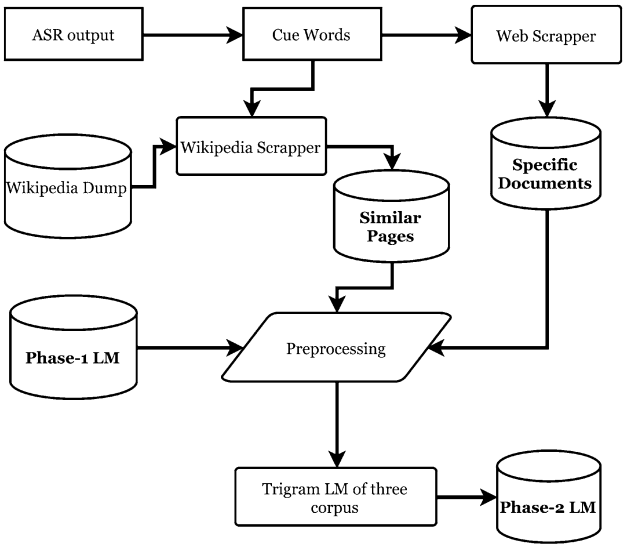

Figure 2. Phase-2 approach for LM generation.

where λ is the interpolation weight between two models, $0 \leq \lambda \leq 1$. Hence, $P_{bw}(w|h)$ becomes our LM that is used during the P1 process.

3.4 *P2*

In this section, we define the P1 process by adapting LMs. The overall framework regarding the P2 approach is presented in figure 2. The figure depicts the steps followed during the P2 process and then bootstrapping those processes with the preceding output of the system. The whole process like searching the cue words, web scrapping and Wikipedia scrapping is automatically performed and later interpolation of these corpora for creating the LM. In pursuance of using a static adaptation process for the creation of a domain-specific LM, the LM is dynamically adapted at the time of recognition and this process is repeated for each topic. All the procedures performed in this stage are explained in further segments.

3.4a *Cue words*: Cue words are the significant words captured from the output produced by the ASR system that may indicate the domain of the lecture speech. The word level information could be gathered by POS tags and NER. Etymologically in the written script, POS tagging and NER of IE are similar to that of American/British English; the major difference lies in the dialect and accent aspects in the spoken language of IE, which makes IE speech recognition a distinct task. For this work, we have used POS tagging for retrieving the cue words relevant to a topic. For POS tagging, we have used Natural Language Toolkit (NLTK) with Python [25]. After analysing all the POS tags, we have found that only a few tags are enough to get the required cue words. Hence, we have fixed it to NNS (Noun Plural), NN (Noun Singular), JJ (Adjective), VBN (verb past participle), JJS (Adjective superlative) and VB (Verb Base-form). Other tags such as adverbs, prepositions, determiners, etc. do not contribute to spotting the cue words and only act as noise in retrieving relevant documents. These fixed POS tags are found to be the most effective to get the theme of the topic. The words that contain these tags that are also repeated more than once are taken as the cue words for further processing. The threshold value for cue word retrieval has been set to 10 based on two factors. The first one is that we applied a trial and error process to find the threshold count. We fixed the threshold to 10 as the search was always able to fetch us 10 cue words. The second factor that instigated us to go ahead with "10" as the threshold for cue word retrieval is to reduce the search space complexity while scrapping documents. A small text, for example, with its POS tags is shown in table 1, where the highlighted words are considered to be the cue words. The highlighted words depicted in this example probe us to think that the text can be related to database management topic. From this example, it is evident that an efficient set of cue words can be obtained for document retrieval process.

Table 1. Example of finding the cue words from the ASR output.

Words	PoS tags
the	DT
database	**NN**
requirements	**NNS**
are	VBP
essentially	RB
method	**JJ**
what	WP
are	VBP
the	DT
data	**NN**
elements	**NNS**

3.4b *Web scrapper*: The cue words obtained as mentioned in section 3.4a are sent as a query to web. Evidently, world wide web consists of a lot more textual information. The amount of information related to any topic/domain and language is increasing very steeply in the web. This aids the retrieval of relevant documents from the web on a particular topic/domain given a set of cue words for scrapping. Based on the Google web scrapper model's ranking algorithm, the first 20 URLs are taken into consideration due to their close proximity towards the theme of the search topic; 20 URLs are obtained for each topic by considering all cue words together while searching. Web scrapping has been implemented using Python's google package,[1] which requires two input parameters: one is the number of documents to be retrieved (20 URLs) and second is the cue

[1] http://pythonhosted.org/google/.

words list (10 cue words). Given these two inputs, the scrapper fetches the matching documents via the search engine from the internet each time. All the texts consisting of URLs are then scrapped and if there exists a URL for a pdf document then using the Python library pdfminer,[2] it is also converted into plain text. This corpus is proximate to the theme required by that particular domain.

3.4c *Wikipedia scrapper*: The English Wikipedia has a huge number of documents/articles on numerous different topics/domains coverage. This data source is available free of cost and also gets updated continuously and dynamically. The downloaded, compressed form of Wikipedia dump[3] is around 10Gb, which when extracted results in a single data file of 40Gb size. All the Wikipages are present in this single file itself; however, searching the required pages from a single file that is 40Gb in size each time will be difficult. Hence, we applied preprocessing (splitting Wiki pages, segmenting and cleaning) on this file and stored the Wikipages into separate files. The files of size <1kB were removed with the intention to reduce the search space and time as well as to remove files with less text content. The set of cue words obtained by the procedure explained in section 3.4a has been used for retrieving the documents from Wikipedia corpus. Here we have used a threshold of 10 cue words that should be present in a particular document and only such documents are retrieved from the Wikipedia corpus. The documents retrieved from this are considered similar to the topic expected and taken for further processing. The number of documents retrieved ranges from 7 to 1100 depending on the presence of minimum 10 cue words in each document per topic. All these retrieved documents are used to build the adapted LM.

3.4d *Adapted LM*: The LM adaptation is performed here in an unsupervised way. It takes the relevant corpus procured from the Wikipedia scrapper and the web scrapper with the assistance of the cue words, which are obtained using the procedure followed in section 3.4a for creating the LM. Hence, our adapted LM consists of the specific data from web scrapper and similar pages from the Wikipedia. These two corpora are preprocessed using the process explained in section 3.2. These preprocessed corpora are then formed into trigram LMs. The P1 LM works like a generic LM used together with other topic-specific LMs. These LMs and the P1 LM are then utilized after forming a combined LM. This combined LM is used after P1 process for recognition of the lecture. This process is bootstrapped as the topics relevant to the data are obtained. Improved recognition from adapting and bootstrapping the LM would be obtained.

Let $P(w|h)$ be the probability, where w is the word and h is the history of the sequence of words. Here, let $P_{bw}(w|h)$ be the P1 LM, $P_g(w|h)$ be the web-specific LM and $P_{ws}(w|h)$ be the Wikipedia document-specific LM used during the P2 and bootstrapping process.

$$P_a(w|h) = \lambda_{bw}P_{bw}(w|h) + \lambda_g P_g(w|h) + \lambda_{ws}P_{ws}(w|h) \quad (7)$$

In Eq. (7), λ is the interpolation weight between two models, $0 \leq \lambda \leq 1$. This $P_a(w|h)$ becomes our combined adapted LM, which incorporates changes according to the topic.

4. Experimental evaluation

In this section, initially, we describe the corpus and data used for training and evaluation. Next, the experimental set-up used for the P1 and P2 evaluations is described in this section. It comprises results that are generated by P1 and P2 processes with bootstrapping. Finally, a comparison of the LMs devised from the proposed approach and the existing LMs is drawn.

4.1 *Data and experimental set-up*

NPTEL[4] lecture videos have been used for building the acoustic models. The video lectures contain various topics from science and engineering lectures at IITs and other premier institutes. These speakers are from various regions of India and they have spoken various accents of IE. We have considered 75 speakers' lecture videos for transcribing the speech in order to train the acoustic model. The data have been video recorded at 44 kHz sampling frequency. These recorded data are then converted into wav format by down-sampling it to 16 kHz and 16-bit mono-file format. Then, we have manually transcribed the audio files in the training data set. A minimum of 15 min of speech of each speaker has been taken into consideration for transcription. The total speech data for training comprise 23 h. The test data consist of 20 min audio each for 10 different domains/topics of NPTEL video lectures. In our experiments, we have considered 10 different domains such as Product Life cycle (PL), Population Studies (PS), Computer Organization (CO), Database (DB), Computer Architecture (CA), Computational Techniques (CT), Scalar Random variables (SR), Axioms Probability (AP), Enzymes (EZ) and Amino Acids (AA). These 10 domains are not a part of the training set. We have purposely chosen such distinct topics in order to have a rigid and unbiased test set. The total test set comprises 3 h and 20 min (200 min) of speech for the evaluation.

In this present work, we have used SphinxTrain[5] for building the acoustic model. Mel frequency cepstral coefficients and their derivatives have been used as features.

[2] https://pypi.python.org/pypi/pdfminer/.

[3] http://en.wikipedia.org/wiki/Wikipedia:Database.

[4] http://nptel.ac.in.

[5] http://cmusphinx.sourceforge.net/wiki/tutorialam.

Then, we have built tri-state context-dependent HMMs for each phone. After several experiments, we decided to have the number of Gaussians in GMM modelling as 32 and number of Senones[6] for decision tree clustering to be 1000. The LM has been built using a variKN toolkit [26] considering the frequent 64k words for trigram LMs. The interpolation of the LM has been performed using a SRILM toolkit [27]. We have used WER (%), perplexity and out of vocabulary (OOV) rate (%) as the evaluation metrics for analysing the performance of LMs.

4.2 *P1 evaluation*

The performance evaluation of P1 model is carried out using the test data with 10 different domains. In case of P1 approach, the generic LM is used for all the 10 domains. The performance using a generic language model that does not have any prior knowledge about the test set domains is analysed for lecture recognition. The performance of the P1 recognition in terms of WER is shown in figure 3. These results are considered as the baseline results for further comparison and evaluation. The least WER of 19.5% for AA and the highest WER of 54.6% for CA have been obtained. The average WER for P1 has been found to be around 40.08%.

Further, the perplexity (PPL) and OOV rate of the LM from P1 for the words in the test set are given in table 2. It can be noticed from table 2 that AA has the lowest WER since it has the lowest perplexity. CA has the highest WER because of its higher perplexity with high percentage of OOV rate. Hence, we can come to the conclusion that better recognition can be achieved if the LM consists of lower perplexity and OOV rates. From this section, the overall analysis for the P1 process for lecture speech recognition is elucidated.

4.3 *P2 evaluation*

To improve the performance of the lecture speech recognition, we carried out the P2 process. The P2 process comprises web scrapping and Wikipedia scrapping with the help of cue words congregated from the transcriptions results of the preceding phase. It can be noticed from figure 4 that the WER after P1 process decreases for all the topics gradually. The bootstrapping (B1–B3) that has been performed further helps in decreasing the WER. The bootstrapping helps us in getting cue words more similar and closer to the domain of the lecture. Each lecture's domain consists of its own LM that differs depending on the lecture and cue words obtained. The LM created for each lecture topic after passing through all the phases of bootstrapping would differ from one another by following the methodology explained as P2 in section 3.4.

[6]http://cmusphinx.sourceforge.net/wiki/tutorialconcepts.

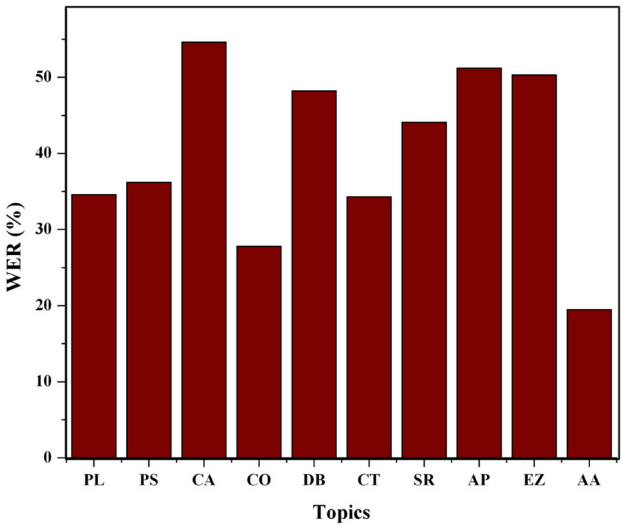

Figure 3. The WER% evaluation from phase 1.

Table 2. The performance evaluation analysis of phase 1.

Topic	WER (%)	Words	PPL	OOV (%)
PL	34.6	2189	291.381	0.41
PS	36.2	2694	378.729	0.67
CA	54.6	2529	267.86	0.28
CO	27.8	2284	52.808	0.04
DB	48.2	3350	278.22	0.27
CT	34.3	2772	188.071	0.4
SR	44.1	2839	232.205	0.11
AP	51.2	2693	208.29	0.04
EZ	50.3	2724	294.32	0.95
AA	19.5	2268	38.96	1.01

The optimal (best) results obtained using the P2 LM with bootstrapping are described in table 3. Here, the results are optimal in the sense that the average WER reduction is negligible after that process. Even though, for the current experiments, the best results are given by P2 LMs, this may not be exactly true for other test datasets. However, the bootstrapping process is similar and the model will deliver more or less similar performance. Even when subjected to any other test set, the results are not distorted much. Table 3 comprises all the evaluation parameters like words, PPL, WER and OOV rate of optimal results. We are able to notice a reduced WER when compared with section 4.2 for the similar topics. The CA, which had the highest WER in P1 of 54.6% has been reduced to 29.3%, showing a difference of approximately 25%. Similarly, we can observe that there is a reduction in the performance based on the WER for each of the topics. The difference in WER varies from topic to topic from 2% to approximately 25%. The reduction in the perplexity for most of the topics is also

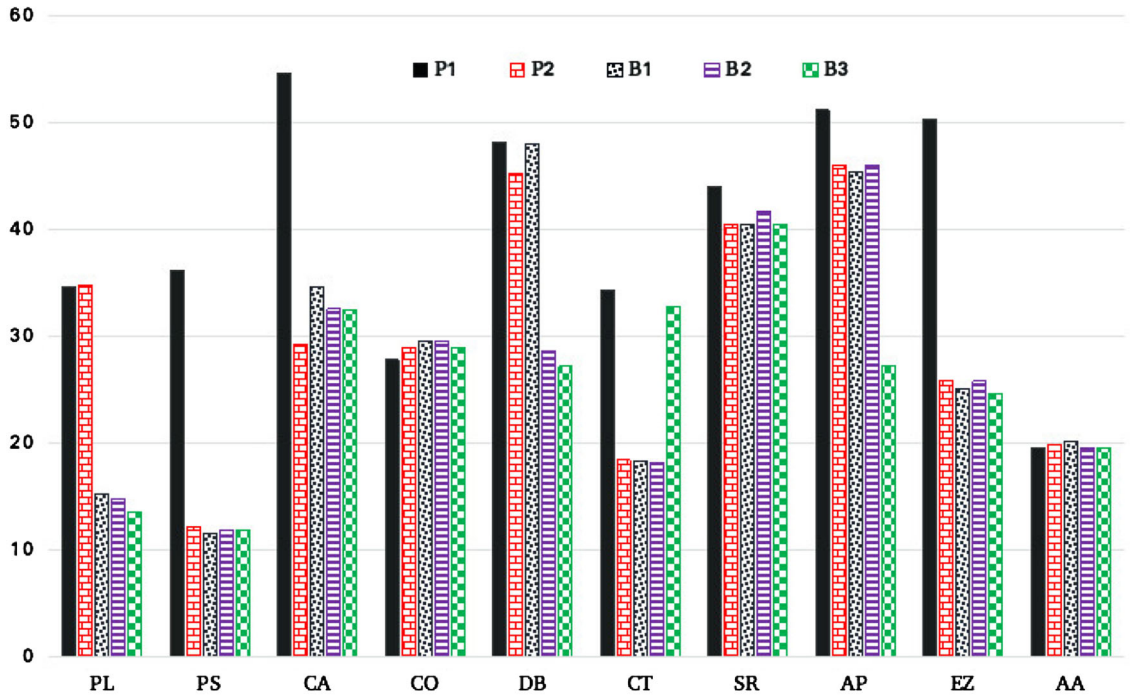

Figure 4. The comparative WER% from phase 1 with phase 2 including bootstrapping.

Table 3. The performance of the optimal results obtained from each topic.

Topic	WER (%)	Words	PPL	OOV (%)
PL	13.6	2189	12.1	0.05
PS	11.6	2694	18.142	0.04
CA	29.3	2589	12.847	0.08
CO	29	2284	66.697	0.04
DB	27.2	3350	17.366	0.06
CT	18.1	2772	13.449	0.04
SR	40.5	2839	133.66	0.07
AP	27.3	2693	19.983	0
EZ	24.7	2724	23.757	0.04
AA	19.5	2268	46.494	0.31

clearly seen. The reduction in the OOV rate on par with the perplexity also can be seen prominently as the OOV rate for one topic is observed to be zero. To achieve the optimal results, the bootstrapping should be applied on the P2 at least once. It has been noticed that the important cue words gets noticed after the P2 process for B1 process, which helps better recognition by reducing WER%. From the P22 output it can be observed that the cue words are more inclined towards the domain under consideration and is depicted in table 4. The extra or filler words gets reduced during the bootstrapping, leading us to relevant cue words. Mostly optimal results are obtained before reaching the B3 phase for LM adaptation.

Table 4. Cue words composed from the output for the domain AP where the highlighted terms are the cue words included that are not present in the previous phase.

From P1	From P2	From B1	From B2
event	event	probability	probability
sample	**probability**	event	event
axiom	numbers	axioms	axioms
space	axiom	numbers	numbers
exclusive	**axioms**	space	space
equals	**development**	development	development
axioms	point	sample	point
states	sample	point	sample
problem	known	conditions	conditions
numbers	space	axiom	axiom
events	**probabilities**	theory	theory
real	**conditions**	important	important
world	**theory**	know	know
equal	**important**	such	known
point	know	known	equal
known	**equal**	equal	real
nothing	**real**	real	

The overall average WER decrease from P1 to P2 with bootstrapping can be seen in figure 5. The decrease from P1 (base) to P2 is approximately 9.96%, which itself is a huge difference in WER without undergoing the bootstrapping process. The bootstrapping further helps in nearly 1.27% reduction from P2 recognition and is represented as B1.

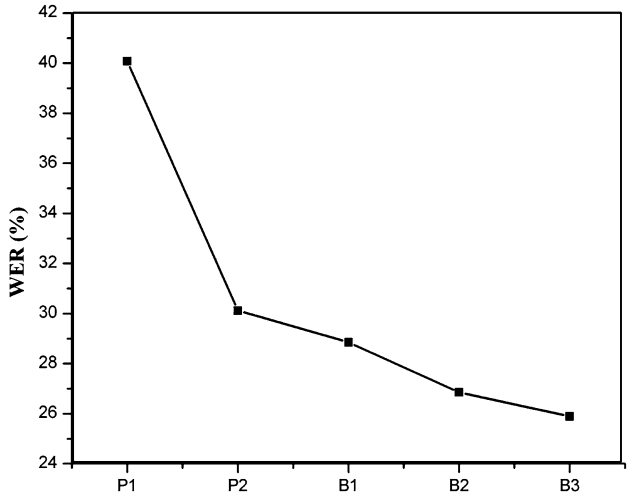

Figure 5. The average WER% from the base phase 1 to phase 2 with bootstrapping.

The WER has further reduced during the B2 phase to about 2% approximately. We have continued the bootstrapping process upto B3 phase for our task which further reduced the WER by nearly 0.96%. We have stopped the bootstrapping process at B3 since the reduction in WER is below 1%. For some topics, the WER is similar to the previous values and for some it is different. Sometimes there is an increase in the WERs which can be due to wrong identification of the cue words, which in-turn affects the document retrieval from web and Wikipedia dump. Also, it is clear that least WER can be achieved within two steps of bootstrapping to avoid further processing of the bootstrapping process for LM adaptation.

4.4 *Comparison to existing LMs*

In this section, a comparison of the LMs devised by proposed framework to the existing LMs has been performed. The existing LMs considered for the comparison are HUB-4 LM [28], Gigaword corpus LM and the CMU Sphinx LM [29]. They are trigram LMs of vocabulary size 64k words. The selection of these LMs has been carried out on the basis of literature, as these models are widely used for large vocabulary speech recognition [30–34]. A description of LMs is given here.

Gigaword LM: This LM is trained using the newswire text provided in the English Gigaword corpus.[7] It consists of trigram LM, created by 64k words with non-verbalized punctuation (NVP).

HUB-4 LM: This LM is built from the most frequent 64k words in the Broadcast News. A trigram model with Katz smoothing is trained on the 64k words of Broadcast News training data.[8]

CMU LM: The CMUSphinx[9] has created a generic LM coverage from web text.

The comparison of the models is done using the standard metrics such as WER (%), perplexity (PPL) and OOV rate (%). In addition to the standard metrics, we added WER recovery and absolute difference metrics. The WER recovery [9] is based on three forms: first is WER_I, which is the initial WER (CMU+Giga), second is WER_S, which is the sustituted/checked WER (P1, P2, B1 and B2) and third is WER_A, which is the least WER achieved (B3). It gives

$$WER\ recovery = \frac{WER_I - WER_S}{WER_I - WER_A} \times 100\%.$$

The absolute difference in WER depicts the value of the WER decrease or increase from the preceding value of WER. It can be denoted as

$$absolute\ difference = WER_P - WER_C,$$

where WER_P is the preceding WER and WER_C is the WER from the current LM.

The LMs considered for evaluation are Gigaword LM (GIGA), CMU Sphinx LM (CMU), HUB-4 LM (HUB4), P1, P2 and P2 with bootstrapping (B1–B3). The corresponding results showing the variation in terms of OOV and perplexity are illustrated in table 5. From table 5, it can be realized that our LM's performance surpassed those of the existing LMs. The perplexity and OOV rate of P1 LM itself attained way better results than those of the existing LMs. The least PPL and OOV rate was noticed in B2 to be 48.55% and 0.073%, respectively. This clearly justifies that the proposed approach for language modelling works superior to the existing LMs.

Further, we analysed the models based on the recognition performance of the models using WER, WER recovery and absolute difference in WER as depicted in table 6. Table 6 shows that the WER of proposed approach is below 40% and the available models are above 47% WER. The maximum of 40.08% WER is noticed during P1 phase and the minimum WER is achieved during B3 phase for language modelling, which is around 25.89%. For the WER recovery, the initial WER was fixed as 47.12%, which is achieved with the combination of CMU and Giga LMs to give a greater coverage of words. The target WER is 25.89%, which is the least WER from B3 phase. Here, it can be noticed that at each phase there is a decrease in the WER and it in turn marks an increase in performance at

[7] https://www.keithv.com/software/giga/.

[8] http://www.itl.nist.gov/iad/mig/publications/proceedings/darpa97/html/seymore1/seymore1.htm.

[9] http://cmusphinx.sourceforge.net/wiki/tutoriallmadvanced.

Table 5. PPL and OOV(%) of existing and proposed two-phase approach LMs.

	GIGA	HUB4	CMU	CMU+ GIGA	P1	P2	B1	B2	B3
PPL	445.62	390.16	459.07	396.66	222.08	84.57	78.61	**48.55**	52.47
OOV (%)	3.104	1.6102	1.357	1.618	0.418	0.114	0.11	**0.073**	0.084

Table 6. The WER(%), WER recovery (%) and the absolute difference in WER results achieved by existing, and combination of existing and the proposed two-phase LM with bootstrapping.

Language models	WER	WER recovery (%)	Absolute difference
GIGA	49.13	–	
HUB4	48.36	–	
CMU	48.35	–	
CMU + GIGA	47.12	–	
P1	40.08	33.16	7.04
P2	30.12	80.07	9.96
B1	28.85	86.06	1.27
B2	**26.85**	95.48	2.00
B3	**25.89**	–	0.96

each phase level including bootstrapping. The absolute difference in the WER depicts the amount of decrease in the WER that is achieved from the existing LMs to the proposed approach. The least WER is achieved by CMU + GIGA combination of LMs from the existing LM. The WER difference achieved during the P1 is 7% from CMU + GIGA, P2 is 9% from P1, B1 is 1% from P2, B2 is 2% from B1 and B3 is 0.9% from B2. From table 6, it can be clearly noted that our proposed model achieves greater reduction in WER and performs better than the existing LMs. This confirms and validates that the proposed two-phase methodology of LM is a striking approach for lecture recognition task. This model can be used for automatic transcription of video/audio lectures. The mundane, highly time-consuming manual transcription process of video/audio can be avoided. People with hearing disabilities will also be the benefactors of this model as the audio in lectures will be available for them in text format to read. Other applications of this model include keyword identification and other transcription-oriented tasks.

5. Conclusion

In this paper, we have experimented the automatic transcriptions of lectures in IE. We have proposed two-phase architecture with bootstrapping to adapt the LM in a way such that the LM is in close proximity to the topic spoken in the lecture speech. Here, we identify the topic-specific cue words linguistically and then the relative documents have been incorporated in building the LM at each step. Using this two-phase architecture, the performance of the lecture recognition task has been improved gradually across the topics. The results show that the overall average WER of the system gets reduced approximately by 14.19% and the perplexity of the LM gets decreased considerably from P1. The base WER from P1 LM has been 40.08% and perplexity nearly 222. The reduced WER achieved by the P2 approach after bootstrapping is 25.89% and perplexity reduced to about 48.55. The performance of the proposed approach can be witnessed clearly with a drop in WER as well as perplexity. If the WER of the base model is below 20%, then even though we observed a decrease in perplexity and OOV, we could not observe a significant reduction in WER. One possible reason could be irrelevant or additional cue words identification by our approach, which is in turn culpable/responsible for topic-specific corpus generation. The performance of the proposed approach outperforms those of existing LMs with respect to each of the evaluation metrics such as perplexity, OOV rate and WER. This confirms and validates that the proposed two-phase methodology of LM is a beneficial and more preferable approach for lecture recognition task.

References

[1] Krishnan M S 2009 NPTEL: A programme for free online and open engineering and science education. In: *IEEE International Workshop on Technology for Education T4E'09*, Bangalore, India, pp. 1–5

[2] Wells J C 1982 Accents of English, Vol. 1. Cambridge University Press, USA

[3] Murthy K N and Kumar G B 2006 Language identification from small text samples*. *J. Quant. Linguist.* 13: 57–80

[4] Wölfel M 2009 *Robust automatic transcription of lectures*. PHD Thesis, Universitätsverlag Karlsruhe, Karlsruhe

[5] Jurafsky D and Martin J H 2000 *Speech & language processing*. Pearson Education, India.

[6] Phull D K and Kumar G B 2016 Investigation of Indian English speech recognition using CMU Sphinx. *Int. J. Appl. Eng. Res.* 11: 4167–4174

[7] Echeverry-Correa J D, Ferreiros-López J, Coucheiro-Limeres A, Córdoba R and Montero J M 2015 Topic identification techniques applied to dynamic language model adaptation for automatic speech recognition. *Expert Syst. Appl.* 42: 101–112

[8] Watanabe S, Iwata T, Hori T, Sako A and Ariki Y 2011 Topic tracking language model for speech recognition. *Comput. Speech Lang.* 25: 440–461

[9] Novotney S, Schwartz R and Khudanpur S 2016 Getting more from automatic transcripts for semi-supervised language modeling. *Comput. Speech Lang.* 36: 93–109

[10] Oger S and Linarès G 2014 Web-based possibilistic language models for automatic speech recognition. *Comput. Speech Lang.* 28: 923–939

[11] Chen B and Chen K Y 2013 Leveraging relevance cues for language modeling in speech recognition. *Inf. Process. Manag.* 49: 807–816

[12] Eickhoff C and de Vries A P 2016 Robust statistical methods in web retrieval. *ACM SIGWEB Newsletter* p. 4

[13] Brychcín T and Konopík M 2015 Latent semantics in language models. *Comput. Speech Lang.* 33: 88–108

[14] Haidar M A and O'Shaughnessy D 2015 Unsupervised language model adaptation using lda-based mixture models and latent semantic marginals. *Comput. Speech Lang.* 29: 20–31

[15] Toral A, Pecina P, Wang L and van Genabith J 2015 Linguistically-augmented perplexity-based data selection for language models. *Comput. Speech Lang.* 32: 11–26

[16] Karpov A, Markov K, Kipyatkova I, Vazhenina D and Ronzhin A 2014 Large vocabulary Russian speech recognition using syntactico-statistical language modeling. *Speech Commun.* 56: 213–228

[17] Liu X, Gales M J and Woodland P C 2014 Paraphrastic language models. *Comput. Speech Lang.* 28: 1298–1316

[18] Liu X, Gales M J and Woodland P C 2013 Language model cross adaptation for lvcsr system combination. *Comput. Speech Lang.* 27: 928–942

[19] Munteanu C, Penn G and Baecker R 2007 Web-based language modelling for automatic lecture transcription. In: *Eighth Annual Conference of the International Speech Communication Association*, Antwerp, Belgium, pp. 2353–2356

[20] Sethy A, Georgiou P G and Narayanan S 2005 Building topic specific language models from webdata using competitive models. In: *Ninth European Conference on Speech Communication and Technology*, Lisbon, Portugal, pp. 1293–1296

[21] Yamazaki H, Iwano K, Shinoda K, Furui S and Yokota H 2007 Dynamic language model adaptation using presentation slides for lecture speech recognition. *Proceedings INTERSPEECH 2007*, pp. 2349–2352

[22] Kneser R and Ney H 1995 Improved backing-off for m-gram language modeling. In: *International Conference on Acoustics, Speech, and Signal Processing*, Detroit, MI, USA, Vol. 1, pp. 181–184

[23] Chen S F and Goodman J 1999 An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* 13: 359–393

[24] Goldberg Y 2017 Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, USA., Vol. 10, No. 1, pp. 1–309

[25] Bird S and Loper E 2006 NLTK: the natural language toolkit. In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Association for Computational Linguistics, Philadelphia, Pennsylvania, pp. 63–70

[26] Siivola V, Creutz M and Kurimo M 2007 Morfessor and VariKN machine learning tools for speech and language technology. In: *Eighth Annual Conference of the International Speech Communication Association*, ISCA, Antwerp, Belgium, pp. 1549–1552

[27] Stolcke A 2002 SRILM-an extensible language modeling toolkit. In: *Seventh international conference on spoken language processing*, Denver, Colorado, USA, pp. 901–904

[28] Schalkwyk Y Y X W J and Cole R 1998 Development of CSLU LVCSR: the 1997 darpa hub4 evaluation system. *Complexity* 24: 7–27

[29] Seymore K, Chen S, Doh S, Eskenazi M, Gouvea E, Raj B, Ravishankar M, Rosenfeld R, Siegler M, Stern R and Thayer E 1998 The 1997 CMU Sphinx-3 English broadcast news transcription system. In: *DARPA Broadcast News Transcription and Understanding Workshop*, Pittsburgh, PA, USA, p 5

[30] Wiesler S, Irie K, Tüske Z, Schlüter R and Ney H 2014 The RWTH English lecture recognition system. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, pp. 3286–3290

[31] Glass J, Hazen T J, Hetherington L and Wang C 2004 Analysis and processing of lecture audio data: Preliminary investigations. In: *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 9–12

[32] Park A, Hazen T J and Glass J R 2005 Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, Philadelphia, PA, USA, Vol. 1, pp. 497–500

[33] Kim W and Khudanpur S 2004 Cross-lingual latent semantic analysis for language modeling. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, Montreal, Que., Canada, Vol. 1, pp. I–257

[34] Liu X, Gales M J F and Woodland P C 2013 Use of contexts in language model interpolation and adaptation. *Comput. Speech Lang.* 27: 301–321