Learning a Language Model from Continuous Speech

Graham Neubig, Masato Mimura, Shinsuke Mori, Tatsuya Kawahara

Graduate School of Informatics, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan

Abstract

This paper presents a new approach to language model construction, learning a language model not from text, but directly from continuous speech. A phoneme lattice is created using acoustic model scores, and Bayesian techniques are used to robustly learn a language model from this noisy input. A novel sampling technique is devised that allows for the integrated learning of word boundaries and an n-gram language model with no prior linguistic knowledge. The proposed techniques were used to learn a language model directly from continuous, potentially large-vocabulary speech. This language model was able to significantly reduce the ASR phoneme error rate over a separate set of test data, and the proposed lattice processing and lexical acquisition techniques were found to be important factors in this improvement.

Index Terms: language acquisition, word segmentation, Pitman-Yor language model, Bayesian learning

1. Introduction

Language models are an important element of automatic speech recognition (ASR) systems, as they provide linguistic constraints to resolve the acoustic ambiguity inherent in continuous speech. Traditionally language models are trained from text, preferably in the domain and style of the speech to be recognized.

However, learning language from text is quite different from human language learning, which is performed primarily on speech or other sensory data [1]. In addition, there are thousands of languages or dialects in the world for which there is very little or no digitized text, or even no official writing system. There is also often a large disconnect between written and spoken speech, due to the presence of disfluencies or colloquial expressions [2]. For all of these reasons, it is of both theoretical and practical interest to examine the acquisition of models of language not from text, but directly from recorded speech.

This paper presents a novel approach to unsupervised learning of a lexicon and a language model directly from audio recordings of continuous speech. Specifically, we adopt a technique for word segmentation using the Hierarchical Pitman-Yor language model [3], applying it to phoneme lattices generated without any linguistic information. Weighted finite state transducers (WFSTs) are used to compose the phoneme lattices with the language model, and Gibbs sampling over the composed lattice is used to perform Bayesian inference.

A language model learning experiment is conducted using meeting speech with a potentially large vocabulary, and the effectiveness of the model is evaluated with phoneme error rate (PER). We investigate the effect of jointly learning the lexicon and the language model, and also the effect of using phoneme lattices to mitigate the effect of phoneme recognition errors in the training process.

2. Unsupervised Word Segmentation

Unsupervised word segmentation has been studied for many years, particularly as a means to elucidate the process of language acquisition in human infants. In recent years, findings that infants are able to use statistical cues to determine word boundaries [4] has spurred a particular interest in statistical methods for unsupervised word segmentation [5, 3, 6]. The majority of these methods are evaluated on text, either of verbatim phonetic transcriptions with word spaces removed, or on word segmentation for languages such as Chinese or Japanese that lack explicit boundaries between words.

In general, these methods assume that an observed corpus $\mathcal{X}$ consists of a number of strings $\mathbf{x} = x_1, \ldots, x_I$, each of which has been generated by a language model G. A prior probability over the space of possible models $P(G)$ is specified, and maximum a posteriori (MAP) or Bayesian inference are used to find models with high joint probability

$$P(\mathcal{X}, G) = P(\mathcal{X}|G)P(G).$$

This paper expands on a method presented by Mochihashi et al. [3] that uses the Hierarchical Pitman-Yor language model (HPYLM) to perform word segmentation. The HPYLM is based on a stochastic process called the Pitman-Yor process, which allows elegant handling of smoothing in a Bayesian context, and performs similarly to state-of-the-art heuristic smoothing techniques such as Kneser-Ney [7]. An HPYLM LM of order n has three sets of hyperparameters, a base measure LM_0, which defines the LMs vocabulary and a prior probability over each word in the vocabulary, as well as a set of discounts $\mathbf{d}_1^n$ and strengths θ_1^n, which define the degree of smoothing to be done at each history length

$$LM \sim HPY(LM_0, \mathbf{d}_1^n, \theta_1^n). \qquad (1)$$

In Mochihashi et al.'s method, it is assumed that the corpus consists of independent character strings, each generated independently by an HPYLM. LM generates not the character sequence $\mathbf{x}$, but a word sequence $\mathbf{w}$, which results in the character sequence when the characters of each word are concatenated (indicated by the function $ct(\mathbf{w})$). Thus, $P(\mathcal{X}|LM)$ can be modeled according to the following equation:

$$P(\mathcal{X}|LM) = \prod_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}|LM)$$
$$= \prod_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{w} \in \{\tilde{\mathbf{w}}:ct(\tilde{\mathbf{w}})=\mathbf{x}\}} P(\mathbf{w}|LM).$$

This language model is generated by a hierarchical Pitman-Yor process, as described in Equation (1). While LM is a model over words, it is necessary to model the relationship between each word and its constituent characters. This is done through
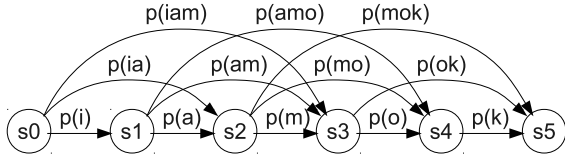
Figure 1: *A WFSA representing a unigram segmentation (words of length greater than three are not displayed).*

the use of a "spelling model" SM, another HPYLM over characters. SM is used as a base measure for LM, and a uniform distribution U over all characters in the corpus is used as a base measure for SM, resulting in the following generative model for the entire corpus:

$$SM \sim HPY(U, \mathbf{d}_{SM}, \theta_{SM})$$
$$LM \sim HPY(SM, \mathbf{d}_{LM}, \theta_{LM})$$
$$\mathcal{X} \sim LM.$$

LM and SM together will be designated as G (for "grammar").

As the main objective of a language model is generally to assign a probability to an unseen character string $\mathbf{x}$, we are interested in calculating the predictive distribution

$$P(\mathbf{x}|\mathcal{X}) = \int_G \sum_{\mathbf{w} \in \{\tilde{\mathbf{w}}:ct(\tilde{\mathbf{w}})=\mathbf{x}\}} P(\mathbf{w}|G)P(G|\mathcal{X})dG.$$

However, computing this function directly is generally computationally difficult, or at least intensive. To reduce this computational load, the Viterbi approximation can be made, assuming that the probability of $\mathbf{x}$ is equal to that of its most likely segmentation. Gibbs sampling is used to take S samples of G, according to the posterior distribution $P(G|\mathcal{X})$, and the mean of $P(\mathbf{w}|G_s)$ is used to approximate the true posterior distribution

$$P(\mathbf{x}|\mathcal{X}) \approx \frac{1}{S}\sum_{s=1}^{S} \max_{\mathbf{w} \in \{\tilde{\mathbf{w}}:ct(\tilde{\mathbf{w}})=\mathbf{x}\}} P(\mathbf{w}|G_s). \quad (2)$$

Mochihashi et al. introduced a blocked Gibbs sampling method that samples segmentations sentence-by-sentence based on the posterior distribution $P(G|\mathcal{X})$. This is done using a technique called *forward-filtering and backward-sampling*, a concept similar to that of the forward-backward algorithm for hidden Markov models (HMM).

Here, we present an intuitive explanation of this algorithm, in which all segmentation candidates for a particular character sequence are represented as a weighted finite state machine (WFSM). An example of a weighted finite state acceptor (WFSA) for the unigram segmentation model of "i am ok" can be found in Figure 1. In the forward filtering step, forward probabilities are pushed from the start state to following states in the acceptor. Given the start state s_0, which has a forward probability f_0 equal to 1, the weights of the following states are updated as follows:

$$f_1 = P(\text{i}) * f_0$$
$$f_2 = P(\text{ia}) * f_0 + P(\text{a}) * f_1$$
$$\vdots$$

In the backward sampling step, a path through the WFSA is sampled according to forward probabilities and transition probabilities for each edge. For example, the edge incoming to state s_5 is sampled according to

$$P(s_4 \rightarrow s_5) = P(\text{k}) * f_4$$
$$P(s_3 \rightarrow s_5) = P(\text{ok}) * f_3$$
$$\vdots$$

Through this process, a segmentation of the character distribution can be accurately sampled from the posterior probability $P(\mathbf{w}|G)$. While the example shows a unigram distribution, the same technique can easily be used for higher-order n-grams by transitioning to different states based on the n-gram history. In fact, this sampling method can be applied to any non-cyclic probabilistic WFSM, a fact which we will use in the following section to sample from phoneme lattices.

3. Learning a Language Model from Speech

Despite the relatively active research on unsupervised word segmentation on text, there has been less work on word segmentation or lexical discovery on actual speech. Some of the few examples include attempts to apply probabilistic models described in the previous section to one-best phoneme recognition results [5, 8]. Driesen et al. [9] apply a unigram multigram model using maximum likelihood estimation and heuristic cutoff criteria to word discovery on speech lattices for digit recognition. In addition, there has been work using cross-channel information to disambiguate boundaries [1], or apply audio matching methods to search for similar segments in raw speech, which may represent phonemes, words, or multi-word phrases [10, 11].

The method presented here differs from these methods in two ways. First, in contrast to previous research using one-best recognition results, we learn over a phoneme lattice annotated with acoustic model scores, allowing the model to absorb some of the noise inherent in phoneme recognition. Second, in contrast to previous research recognizing isolated words, this model learns a rudimentary syntax in the form of an n-gram language model. This language model can be used for speech recognition on data outside of the training set. An additional motivation for using context is that it has been proven essential to acquire word boundaries that match with human intuition [6].

As mentioned in the previous section, the method of forward filtering and backward sampling can be applied to any non-cyclic WFSM with probabilistic weights. Operations on weighted finite state transducers (WFSTs, [12]) provide a natural way to create a WFSM for sampling over phoneme lattices. In particular, for each sampling operation, we perform the composition of three transducers.

- X: A WFSM representation of the phoneme lattice with acoustic model scores.

- L: A WFST representing the lexicon, which transduces $\mathbf{x}$ into all possible segmentations $\mathbf{w}$.

- G: A WFSA that accepts all possible $\mathbf{w}$ and assigns the probability $P(\mathbf{w}|G)$.

The construction of X is straightforward, and methods for construction of L and G are described in [12]. However, here G must represent two language models, LM and SM, which can be expressed in WFST format as shown in Figure 2. The key to the representation is the appropriately weighted edges falling back from the base state of LM to SM, and the edges accepting
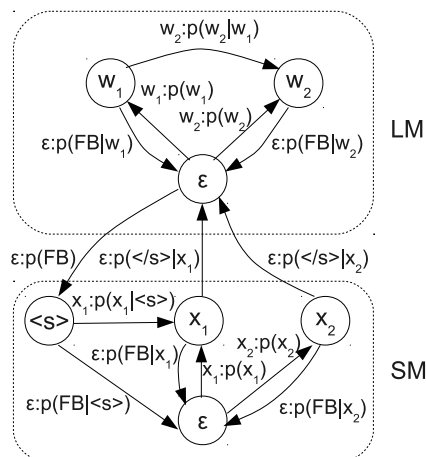
Figure 2: *An example of the WFST for G. "FB" indicates a fall-back to a shorter history, while "<s>" and "</s>" indicate the start and terminal symbols of the SM respectively.*

the terminal symbol for unknown words transitioning from SM to the base state of LM.

Through the composition of the three transducers $X \circ L \circ G$, it is possible to create a WFSA where each path represents a possible segmentation weighted with its posterior probability given G and the scores of the acoustic model AM:

$$P(\mathbf{x}|G) = P(\mathbf{x}|\mathbf{w}; AM)P(\mathbf{w}|G).$$

By performing forward filtering and backward sampling over this lattice, it is possible to obtain a single segmented phoneme string. Other than the difference between sampling over a phoneme lattice and sampling over a phoneme string, the algorithm described in [3] can be used without any modification. This allows for the unsupervised learning of an HPYLM over a corpus of phoneme lattices, resulting in a language model and a lexicon consisting of phoneme strings.

4. Experimental Results

We tested the feasibility of the proposed method on continuous speech from meetings of the Diet (national congress of Japan). This was chosen as an example of speech with a potentially large vocabulary, as opposed to infant-directed or small-vocabulary speech used in some previous works [1, 9].

4.1. Experimental Setup

A triphone acoustic model was used to create phoneme lattices from meetings of the Japanese Diet[1]. Decoding was performed using a language model that provided a uniform distribution over 385 syllables, which exhaustively represent the majority of the transitions allowed by the triphone model[2].

Language models were trained using data sizes varying from 119 to 1904 utterances (7.9 and 116.7 minutes respectively). 500 utterances (27.2 minutes of speech) were held out

---

[1]This dependence on an acoustic model indicates that this is not an entirely unsupervised method. However, some work has been done on unsupervised or language-independent acoustic model training [13], which is another difficult challenge not covered in this work.

[2]Syllable-based decoding was necessary due to the limits of the decoding process, and is not a fundamental part of the proposed method. Phoneme-based decoding will be examined in the future.
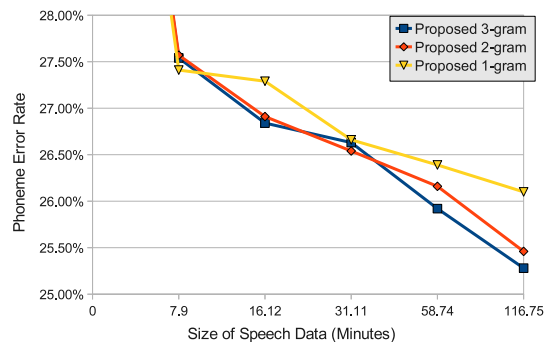

Figure 3: *Phoneme error rate by model order.*

as a test set. As a measure of the quality of the language model learned by the training process, we used phoneme error rate when the language model was used to search the phoneme lattice of the held-out set. Phoneme error rate is an objective measure of how well the learned model is modeling the language, as opposed to word segmentation accuracy or lexicon accuracy, which may depend heavily on a particular segmentation standard. The phoneme error rate given no linguistic information (a zero-gram language model) was 34.20%. The oracle phoneme error rate of the phoneme lattice on the test set was 8.10%, indicating that even given a perfect language model a fair amount of noise would remain in the results.

Fifty samples of G were taken after twenty iterations of burn-in, the first ten of which were annealed according to the technique presented by Goldwater et al. [6]. A language model scaling factor was used, which was set to five, with values between five and ten producing similar results in preliminary tests. While Equation (2) approximates the probability using the average maximum-segmentation probability of the language models, search for such a solution when word boundaries vary from language model to language model is a non-trivial problem. As an approximation to this, the one-best solution was found for each of the sampled language models, and the fifty separate solutions were combined together using ROVER [14].

4.2. n-gram Context Dependency

In the first experiment, the effect of using context information in the learning process was examined. A language model was learned using an HPYLM trigram spelling model, and setting the n of the HPYLM language model to 1, 2, or 3. The results with regards to phoneme error rate can be found in Figure 3.

First, it can be seen that a language model learned directly from speech was able to improve the accuracy by 7% absolute PER or more compared with when no linguistic information was used. This is true even with only 7.9 minutes of training speech. In addition, the results show that the bigram and trigram models outperform the unigram model, particularly as the size of the training data increases. We were also able to confirm the observation [6] that unigram models tend to undersegment, grouping together multi-word phrases instead of actual words. This is reflected in the vocabulary and n-gram sizes of the three models after the final iteration of the learning process displayed in Table 1. The vocabulary size of the unigram model is much larger than that of the bigram and trigram models as a result of this undersegmentation, with the lack of complexity in the LM being transferred to the SM.

1055

Table 1: *Vocabulary and model size for 116.7 minutes of speech.*

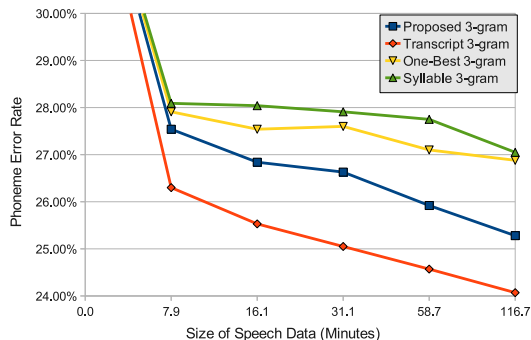|	1-gram	2-gram	3-gram
Vocabulary	4480	1351	708
LM entries	4480	16150	38759
SM entries	9624	3869	2426

Figure 4: *Phoneme error rate for various training methods.*

4.3. Comparison with Other Techniques

We also compared the proposed lattice processing method with three other language model construction methods. First, we trained the proposed method not on word lattices, but one-best ASR results. Second, to examine whether estimation of word boundaries is beneficial, we trained a syllable trigram language model on one-best results.

Finally, as an approximate upper bound on the performance of the proposed method, a language model was built using a human-created verbatim transcription of the utterances. Word segmentation and pronunciation annotation were performed with the KyTea toolkit [15], and pronunciations of unknown words were annotated by hand. A trigram model was created on segmented phoneme strings using Kneser-Ney smoothing.

The phoneme error rates for the four methods are shown in Figure 4. It can be seen that the proposed method outperforms the model trained on one-best results, verifying that lattice processing is able to reduce some of the noise inherent in acoustic matching results. It can also be seen that on one-best results, the proposed model outperforms the syllable-based language model for all data sizes. This indicates that it is, in fact, beneficial to acquire lexical units for language modeling.

As expected, the proposed method does not perform as well as the model trained on gold-standard transcriptions. However, it appears to improve at approximately the same rate as more data is added, which is not true for one-best transcriptions. A possible explanation for this gap between lattice processing and the gold standard is the noise introduced by the high oracle error rate (8.10%). By expanding the size of the lattice, or directly integrating the calculation of acoustic scores with sampling, it will be possible to further close this gap.

5. Discussion and Future Work

The results presented here have shown that it is possible to acquire a language model in an unsupervised fashion given only speech and an acoustic model. In particular, the simultaneous acquisition of a lexicon and language model, as well as consideration of multiple hypotheses through lattice processing were shown to be effective in improving ASR accuracy through unsupervised learning.

This work opens up a number of possible directions for future research in a variety of areas. For example, it can be used to discover a lexicon and language model for under-resourced languages with little or no written text. Another promising expansion of the proposed method is semi-supervised learning, which would allow an existing language model to be enhanced with untranscribed speech for adapting language models to new domains, speaking styles, or dialects. The largest remaining technical challenge is computational efficiency. Acquiring a single sample of an utterance takes time approximately equal to the length of the utterance (three seconds for a three second utterance), resulting in a significant computational load for training. This could be ameliorated by converting full search of the phoneme lattices to beam search, or by expanding to multiple cores using parallel sampling.

6. Acknowledgements

7. References

[1] D. Roy and A. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive Science*, vol. 26, no. 1, pp. 113–146, 2002.

[2] G. Neubig, S. Mori, and T. Kawahara, "A WFST-based log-linear framework for speaking-style transformation," in *Proc. InterSpeech2009*, 9 2009, pp. 1495–1498.

[3] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor modeling," in *Proc. ACL09*, 2009.

[4] J. R. Saffran, R. N. Aslin, and E. L. Newport, "Statistical learning by eight-month-old infants," *Science*, vol. 274, no. 5294, pp. 1926–1928, 1996.

[5] C. de Marcken, "The unsupervised acquisition of a lexicon from continuous speech," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1995.

[6] S. Goldwater, T. L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.

[7] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proc. ACL06*, 2006, pp. 985–992.

[8] A. Gorin, D. Petrovska-Delacretaz, G. Riccardi, and J. Wright, "Learning spoken language without transcriptions," in *Proc. ASRU99*, 1999.

[9] J. Driesen and H. V. Hamme, "Improving the Multigram Algorithm by using Lattices as Input," in *Proc. InterSpeech2008*, 2008.

[10] A. Park and J. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio Speech and Language Processing*, vol. 16, no. 1, p. 186, 2008.

[11] L. ten Bosch and B. Cranen, "A computational model for unsupervised word discovery," in *Proc. InterSpeech2007*, 2007, pp. 1481–1484.

[12] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, no. 2, pp. 269–311, 1997.

[13] L. Lamel, J. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training." *Computer Speech and Language*, 2002.

[14] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU97*, 1997.

[15] G. Neubig and S. Mori, "Word-based partial annotation for efficient corpus construction," in *Proc. LREC2010*, 2010.