Building Class-based Language Models with Contextual Statistics

Shuanghu Bai, Haizhou Li, Zhiwei Lin, Baosheng Yuan

Institute of Systems Science, National University of Singapore, Singapore 119597

Abstract

In this paper, novel clustering algorithms are proposed by using the contextual statistics of words for class-based language models. The Minimum Discriminative Information (MDI) is used as a distance measure. Three algorithms are implemented to build bigram language models for a vocabulary of 50,000 words over a corpus of over 200 million words. The computational cost of algorithms and resulting LM perplexity are studied. The comparisons between the MDI algorithm and the Maximum Mutual Information algorithm are also given to demonstrate the effectiveness and the efficiency of the new algorithms. It is shown that the MDI approaches make the tree-building clustering[2] possible with large vocabulary.

1 Introduction

In order to incorporate the statistical language models into a large vocabulary continuous speech recognition system, it is always necessary to partition the whole vocabulary into a number of classes, or Part-of-Speech (POS) categories. Words within a class share similar syntactic and semantic functionalities. A number of algorithms have been proposed. An algorithm, called maximum mutual information (MMI), has been widely adopted to generate bigram language models[1, 2]. To obtain a C classes language model for a vocabulary size of V with MMI approach, the computation is of the order of V^3, where each operation involves a logarithm. To be practical for large vocabulary size V, larger than 5,000, a greedy merge method is also introduced based on the MMI theory, which still requires one order of V^2C operations. The merging or classification is conducted in a sequential manner, therefore, iterations over the given sample data will certainly improve the results. Unfortunately, for a very large vocabulary, say $V > 50,000$, the computation for a single batch does not allow iterations to happen.

In this paper, we will propose a new framework based on minimum discriminative information (MDI) clustering, which requires less computations to enable an iterative sequential minimization. Three algorithms will be discussed which are of order of less than V^2. In section 2, the MDI distance measure will be studied. In section 3, the procedures of algorithms will be constructed. In section 4, the computational cost will also be addressed. Finally, the LM perplexity studies over a 50,000 words vocabulary with 200 million words training corpus are carried out. It is shown that MDI gives similar results to MMI method while dramatically reducing the computations.

2 Minimum discriminative information

The contextual statistics of a word can be easily obtained from a corpus. Here the contextual information $ci\{w\}$ is defined to be the co-occurrence counts or the frequencies of a word with its neighbouring words at distance l. Given a lexicon of V words, in the case of $l = 1$,

$$ci\{w\} = \{cil\{w\}, cir\{w\}\} \tag{1}$$

where

$$cil\{w\} = \{cl(w_1, w)...cl(w_V, w)\} \tag{2}$$

$$cir\{w\} = \{cr(w, w_1)...cr(w, w_V)\} \tag{3}$$

where $cl(w_v, w)$ is left-bigram counts for w and $cr(w, w_v)$ the right-bigram. Let $c(w)$ denote the word count, 1-gram, for w, we have

$$c(w) = \sum_{v=1}^{V} cl(w_v, w) = \sum_{v=1}^{V} cr(w, w_v) \tag{4}$$

The sum count of $ci\{w\}$ components is $2 \times c(w)$.

Now two principles are given to our algorithms. Firstly, the part of speech of a word can be determined by the part of speech of its contextual words. Secondly, words with similar POS functions are merged into the same class. Therefore, the problem becomes how to

define the similarity of two words in terms of their POS functions, or contextual information. Representing the co-occurrence counts $c(w_1, w_2)$ in probabilities

$$p(w_1/w_2) = c(w_1, w_2)/c(w_2) \tag{5}$$

Eq(1) gives

$$\begin{aligned} & pci\{w\} \qquad (6) \\ = \; & \{pl(w_1/w)...pl(w_V/w), pr(w_1/w)...pr(w_V/w)\} \end{aligned}$$

where $\sum_v pl(w_v/w) = 1$ and $\sum_v pr(w_v/w) = 1$.

The discriminative information between two words w_1 and w_2 is given as

$$\begin{aligned} D(w_1, w_2) &= \sum_{v=1}^{V} pl(w_v/w_1) \log \frac{pl(w_v/w_1)}{pl(w_v/w_2)} \\ &+ \sum_{v=1}^{V} pr(w_v/w_1) \log \frac{pr(w_v/w_1)}{pr(w_v/w_2)} \end{aligned} \tag{7}$$

which is also known as Kullback-Liebler distortion measure or relative entropy.

The objective of partitioning the vocabulary is to find a set of centroids $\{o_c\}$ for cells $\{O_c\}$, $c = 1, C$ which give the minimum global discriminative information

$$\begin{aligned} GDI &= \sum_{c=1}^{C} \sum_{i \in O_c} D(w_i, o_c) \qquad (8) \\ &= \sum_{i=1}^{V} \sum_{v=1}^{V} pr(w_v/w_i) \log pl(w_v/w_i) \\ &+ \sum_{i=1}^{V} \sum_{v=1}^{V} pr(w_v/w_i) \log pr(w_v/w_i) \\ &- \sum_{c=1}^{C} \sum_{i \in O_c} \sum_{v=1}^{V} pl(w_v/w_i) \log pl(w_v/o_c) \\ &- \sum_{c=1}^{C} \sum_{i \in O_c} \sum_{v=1}^{V} pr(w_v/w_i) \log pr(w_v/o_c) \\ &= H(w) - R(w) \end{aligned}$$

where, speaking in information theory, $R(w)$ is the bit rate in transmitting source w with symbol o_c and $H(w)$ is the entropy of source w. Therefore, the relative entropy can be interpreted as the error bit rate when transmitting source w with symbol o. It can be easily seen that $H(w)$ is a constant independent of the partitioning. When the global discriminative information is minimized, $R(w)$ is maximized.

Each partition or class O_c is represented by a centroid word o_c which carries the common POS functions for the class. Let us denote Eq.(6) as

$$pci\{w\} = \{p(k/w), k = 1, 2V\} \tag{9}$$

Given class $O_c = \{w_i, i = 1, v_c\}$, the centroid of O_c, $o_c = \{o(k/o_c), k = 1, 2V\}$ can be estimated[3] by using the minimum distance rule. Notice that the Kullback-Liebler distortion is not a symetric measure, we have

$$o(k/o_c) = \frac{1}{v_c} \sum_{i=1}^{v_c} p(k/w_i) \tag{10}$$

$$o(k/o_c) = \frac{\sqrt[v_c]{\prod_{i=1}^{v_c} p(k/w_i)}}{\sum_{k'=1}^{2V} \sqrt[v_c]{\prod_{i=1}^{v_c} p(k'/w_i)}} \tag{11}$$

respectively for the two distance measuring order $D(o, w)$ and $D(w, o)$. Since the words scatter in a discrete space, o_c by Eq(10) might not be a valid word. One can find the pseudo-centroid by looking in the class for the closest word to o_c.

It is noted the left and right contexts here are considered as two *trigger pairs*[4] concerning w at distance 1, it is straightforward to extend distance l to $L > 1$ to accommodate more contextual information by augmenting the Eq(6) with co-occurrence counts of triger pairs at longer distances. For brevity, only $l = 1$ will be dicussed throughout the rest of the paper.

3 Sequential minimization

As previously stated, a word is characterized as a probability array in terms of contextual information. Clustering words becomes a encoding problem in VQ design. Suppose that a sequence of sample data V is to be clustered into C classes. An exhaustive search, that is, tree-building algorithm, for the V samples will take the operations of order of V^3 as described in [2]. It is too expensive to be feasible for large V. A LBG procedure[5] based on iterative improvement in general yield a good clustering, from which much less computational cost is expected, less than order of V^2. Here, the C most frequent words are assigned into C distinct classes as the initial codewords. Then two procedures proceed iteratively until the average distortion GDI is small enough. (1) Classify the sequence of words, which is ordered by decreasing frequency, into a sequence of classes using the minimum distortion rule. (2) Replace the old reproduction codewords for each class by its estimated centroid. Usually it only takes few iterations to achieve a fairly good result.

Algorithm 1

- step 1: start with initial codebook;
- step 2: classify $w_i, i = 1, V$ with Eq.(7);

- step 3: update the codebook with Eq.(10) or Eq.(11);
- step 4: **if** $GDI < t$ **then** stop **else** step 2.

t is a threshold used to terminate the convergent process. By Eq.(7), it is noted that the logarithm can be prestored to reduce the on-line computation. The computation involves one order of V^2, the number of bigrams, logarithm. Actually, only a fraction of bigrams do occur in the corpus among the V^2 possibilities. For example, we only have 8.6×10^6 non-zero bigrams instead of 2.5×10^9.

In Algorithm 1, the codebook is updated after each batch cycle at step 3. To have the codewords on-line adapted to the changing classes, we suggest to update the codeword after each word merging, which introduces additional one order of V^2 logarithm for each batch cycle. It leads to another implementation.

Algorithm 2

- step 1: start with initial codebook;
- step 2: classify w_i, $i = 1, V$ with Eq.(7), update codebook with Eq.(10) or Eq.(11);
- step 3: **if** $i = V$ **then** step 4 **else** $i = i + 1$, step 2;
- step 4: **if** $GDI < t$ **then** stop **else**, step 2;

To further reduce the computation, an algorithm is proposed to get rid of the logarithm. Given a word w_i, one first assumes that all the observed bigrams from the training corpus are of equal probabilities. Then the contextual co-occurrence frequencies of w_i, $p(k/w)$ are set to y_i for occurrence and $\tilde{y_i}$ for absence as a log zero floor, subject to $\sum_k p(k/w) = 1$. By Eq.(7), we have

$$\begin{aligned} D(w_1, w_2) &= N_1 y_1 \log \frac{y_1}{y_2} + N_2 y_1 \log \frac{y_1}{\tilde{y}_2} \\ &\quad + N_3 \tilde{y}_1 \log \frac{\tilde{y}_1}{y_2} + N_4 \tilde{y}_1 \log \frac{\tilde{y}_1}{\tilde{y}_2} \\ &= \alpha N_1 + \beta N_2 + \gamma N_3 + \delta N_4 \end{aligned} \quad (12)$$

where N_1 is the count of $on-on$ pairs between w_1 and w_2, N_2 the $on-off$ pairs, N_3 the $off-on$ pairs and N_4 the $off-off$ pairs. In this case, the distance measure is basically counting the number of pair occurrences. The simplicity of distance measure allows us to have an exhaustive matching over a large space. The algorithm might be used to have a preclassification of a large vocabulary. The idea is to reduce the search space significantly before detailed clustering is conducted.

Algorithm 3

- step 1: create a class for word w_1;
- step 2: find the closest class o to w_i with Eq.(12);
- step 3: **if** $D(w_i, o) < A$ **then** merge w_i to o **else** create a new class for w_i
- step 4: $i = i + 1$, **if** $i < V$ **then** step 2 **else** stop

It is found that Algorithm 1 and 2 merge V words to form a codebook of size C while Algorithm 3 grows codebook size from 1 to C. A threshold A is predefined to decide whether a new class should be formed. The resulting number of classes C depends on this threshold.

4 Algorithm study

The concept of mutual information, taken from information theory, is proposed as a measure of word association. It reflects the strength of relationship between words by comparing their actual co-occurence with the probability that would be expected by chance. Maximizing the average mutual information will lead to class-based language models of lower perplexities. For the classification with exhaustive matching, the updating process for each merge requires a order of V^2 computations. As we have V words to merge, the total computation is of order of V^3.

In this paper, a MDI measure is introduced which aims at minimizing the average discriminative information incurred by the classification. Now let us discuss how the MDI approach works in the order of V^2.

The distance measure in MMI is to find the least loss in average mutual information among all the potential merging pairs (i, j) by evaluating [2]

$$\begin{aligned} I(i,j) &= \sum_{l \neq i,j} p(w_{i+j}, w_l) \log \frac{p(w_{i+j}, w_l)}{pl(w_{i+j}) pr(w_l)} \\ &\quad + \sum_{m \neq i,j} p(w_m, w_{i+j}) \log \frac{p(w_m, w_{i+j})}{pr(w_{i+j}) pl(w_m)} \end{aligned}$$

which involves similar computation as Eq(7) does. In addition to the mutual information evaluation, to complete the merging step , we must update the counts or probabilities for the merged class. The entire update process for one sample, that is, a word or a class, requires something on the order of V^2 computations, logarithms. With MDI approach, the update process is much simpler as given in Eq(10). Therefore, the entire update process for a sample only involves a order of V, and gains one order over MMI approach.

Although we have reduced the computation by one order with the MDI approach, the exhaustive matching is still too expensive to be practical for large vocabulary, for example, $V > 20,000$. The sequential

minimization approach further reduces the computation to a order of $V \times C$ which becomes a practical implementation for very large vocabulary.

The ideas presented in this paper also makes the sequential minimization possible. As stated earlier, a word is characterized by a contextual statistical array, hence, we can easily find a codeword by Eq(10) in each class which is used to represent the POS features of the class. Thanks to the simplicity of the codeword finding mechanism, the cost of merging of words or classes become less expensive. In Algorithm 1, we do not even bother about updating the codewords after merging a sample. The codewords are updated after a batch merging is completed. In Algorithm 2, codewords are updated after merging a sample. It is shown that there is no much difference between the two algorithms.

5 Experiments

The task is to cluster a vocabulary of $50,000$ words into $1,000$ classes. The corpus used to build the LMs, called training corpus, is a collection of newspaper text from People's Daily. It consists of 200 million words. Another corpus of 448 thousand words, which is exclusive of the training corpus, is used as the test corpus.

The standard measure by which language models (LMs) are assessed is by calculating their perplexity using a sample of test data. Table 1 shows the results of two setups for Algorithm 3, where Algorithm 3 is practiced to pre-classify the vocabulary into a class-based vocabulary. In the E1 experiment, parameters in Eq(12) are set to $\alpha = 1.0$, $\delta = 0.01$ and $\beta = \gamma = -0.01$. In E2 experiment, $\alpha = 1.0$, $\delta = 0.05$ and $\beta = \gamma = -0.05$. which means that the relevance of the contextual information are in the order of $on-on$, $off-off$, then $on-off$ and $off-on$.

	Vocabulary Size	word PP	character PP
	50,000 words	389	57
E1	36,000 classes	425	60
E2	22,000 classes	400	60

Table 1: The perplexity report of preclassification by Algorithm 3

In Chinese, character is the minimum unit of word, therefore, character perplexity also reflects how complex the test corpus is. The scaled down vocabulary is then clustered into our targeted $1,000$ classes by Algorithm 1. Table 2 shows that Algorithm 1 gives similar result as MMI greedy merging does. By greedy merging, MMI approach clusters $50,000$ into $1,000$ directly.

$1,000 classes$	word PP	character PP
A1	680	84
MMI	663	80

Table 2: The comparison between Algorithm 1 and MMI greedy merging algorithm[2]

6 Conclusions

A new algorithm is presented in this paper for class-based n-gram classification, which reduces the computation by one order when compared to MMI method[2]. Although only the immediate left and right contexts are considered, the approach is also able to take long distance contextual information into account. A simplified procedure of the algorithm gets rid of the logarithm operations which is shown to be an efficient and effective approach to preclassify a large vocabulary.

In this paper, a procedure of preclassification followed by detailed classification is also proposed in the MDI framework. The resulting bigram language model is now used in our Mandarin continuous speech recognition system.

References

[1] Jelinek. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*. Morgan Kaufmann, 1990.

[2] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–480, December 1992.

[3] H. Li, J.-P. Haton, J. Su, and Y. Gong. Speaker recognition with temperal transition models. In *Proceedings of European Conference on Speech Technology*, Madrid, Spain, 1995.

[4] R. Rosenfeld. A maximum entropy appraoch to adaptive statistical modelling. *Computer Speech and Language*, 10(.):187–228, . 1996.

[5] R. M. Gray. Vector quantization. In Alex Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*. Morgan Kaufmann, 1990.