Language Model Based on Word Clustering

Lichi Yuan

School of Information Technology, Jiangxi University of Finance & Economics, Nanchang 330013, China
yuan_lichi@hotmail.com

Abstract. Category-based statistic language model is an important method to solve the problem of sparse data. But there are two bottlenecks about this model: (1) the problem of word clustering, it is hard to find a suitable clustering method that has good performance and not large amount of computation. (2) class-based method always loses some prediction ability to adapt the text of different domain. The authors try to solve above problems in this paper. This paper presents a definition of word similarity by utilizing mutual information. Based on word similarity, this paper gives the definition of word set similarity. Experiments show that word clustering algorithm based on similarity is better than conventional greedy clustering method in speed and performance. At the same time, this paper presents a new method to create the vari-gram model.

Keywords: Word clustering, statistical language model, Vari-gram language model

1 Introduction

Statistical language modeling[5] (SLM) has been successfully applied to many domains such as speech recognition, machine translation, information retrieval, and spoken language understanding. The dominant technology in SLM is n-gram models. Typically n-gram models are trained on very large corpora. In constructing n-gram models, we always face two problems. First, for a general domain model, large amounts of training data can lead to models that are too large for realistic applications. On the other hand, for specific domains, n-gram models usually suffer from the data sparseness problem, because large amounts of domain-specific data are usually not available.

When n-gram models are used, we can define clusters for similar words in a corpus. We thus augment word-based n-gram models to cluster-based n-gram models. This has been demonstrated as an effective way to handle the data sparseness problem. There are many different clustering algorithms[1], but they can be classified into a few basic types. There are two types of structures produced by clustering algorithms, hierarchical clustering and flat or non- hierarchical clustering. Flat clustering simply consists of a certain number of clusters and the relation between clusters is often undetermined. Most algorithms produce flat clusters and improve them by iterating a reallocation operation that reassigns objects. The tree of a hierarchical clustering can be produced either bottom-up, by starting with the individual objects and grouping the most similar ones, or top-down, whereby one starts with all the objects and divides them into groups so as to maximize within-group similarity. This paper proposes a bottom-up hierarchical clustering algorithm based on similarity.

This paper also discusses the problem of the design of vari–gram[3]. The cluster n-gram model has been demonstrated as an effective way to deal with the data sparseness problem which exists in the word n-gram model, but this method sacrifice some predictable ability. We can improve the system predict performance by enhancing n, because the number of clusters is much less the number of words. However, it also has a few drawbacks, including: exponential growth in parameters as a function of n, which Increases the storage and computer costs of the recognition search; the problem of sparse data for parameter estimation; and the inability to model long distance relationships. To solve this problem, people propose vari-gram language model, in which model, the value of n varies according to the different ability that history words predict for the current word. But the construction of vari-gram model is a very complicated issue. In this paper, an Absolute Weighted Difference method is presented and is used to construct vari-gram model which has good predictable ability.

2 N-gram Models and Performance Evaluation

The classic task of language modeling is to predict the next word given the previous words. The n-gram model is the usual approach. It states the task of predicting the next of word as attempting to estimate the conditional probability.

$$P(w_i \mid w_1,\cdots,w_{i-1}) = P(w_i \mid w_{i-n+1},\cdots,w_{i-1}) \tag{1}$$

An estimate of the probability $P(w_i \mid w_{i-2},w_{i-1})$ is given by Equation (2), called the maximum likelihood estimation (MLE):

$$P(w_i \mid w_{i-2},w_{i-1}) = \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1})} \tag{2}$$

Where $C(w_{i-2},w_{i-1},w_i)$ represents the number of times the sequence w_{i-2},w_{i-1},w_i occurs in training text.

The most common metric for evaluating a language model is *perplexity.* Formally, the word perplexity PP_w of a model is the reciprocal of the geometric average probability assigned by the test set. It is defined[5] as:

$$PP_w = 2^{-\frac{1}{N_w}\sum_{i=1}^{N_w}\log_2 P(w_i|w_{i-2},w_{i-1})} \tag{3}$$

Where N_w is the total number of words in the test set.

3 Word Similarity and Clustering Algorithm based on Word Similarity

Unlike above method, the base that we cluster words is the similarity between words. First, we must find an appropriate word similarity metric. The common corpus-based approach for computing word similarity is based on representing a word (or term) by the set of its **word co-occurrence** statistics. It relies on the assumption[4] that the meaning of words is related to their patterns of co-occurrence with other words in the text.

Assume that the two words w_1 and w_2 are similar, and then we may infer that they have similar mutual information with some other words (Ido Dagan, 1995). Now we define the similarity between words w_1 and w_2 as follow:

$$sim(w_1,w_2) = \frac{\sum_w P(w)[\min(I(w,w_1),I(w,w_2)) + \min(I(w_1,w),I(w_2,w))]}{\sum_w P(w)[\max(I(w,w_1),I(w,w_2)) + \max(I(w_1,w),I(w_2,w))]} \tag{4}$$

Where $I(w_i,w_j)$ represents the point wise mutual information between the two words w_i and w_j .

$$I(w_i,w_j) = \log\frac{p(w_i,w_j)}{p(w_i)p(w_j)} \tag{5}$$

Where $p(w_i)$ and $p(w_j)$ are the probabilities of the events w_1 and w_2 (occurrences of words, in our case) and $p(w_i,w_j)$ is the probability of the joint event (a concurrence pair).

Based on word similarity, the similarity between clusters C_1 and C_2 may be defined as:

$$sim(C_1, C_2) = \frac{\sum_{w_i \in C_1, w \in C_2} C(w_i) C(w_j) sim(w_i, w_j)}{\sum_{w \ \in C_1} C(w_i) \sum_{w_j \in C_2} C(w_j)} \qquad \textbf{(6)}$$

Where $C(w_i)$, $C(w_j)$ represent the number of the words w_i and w_j occur in the corpus. The left similarity and right similarity between clusters may be defined similarly.

The clustering algorithm is as follow:

(a) Compute similarity between words.

(b) Begin with N (N is the number of words in lexicon) clusters, one for each word.

(c) Select the two clusters which have the biggest similarity, and create a new cluster by merging the two clusters together.

(d) Computer the similarity between the new cluster and other cluster.

(e) Check if the termination condition (the value of the biggest similarity between clusters is less than a predetermined threshold, or the desired number of clusters is reached) is meet, if yes, the program is terminated; or go to (c).

4 Vari-gram Language Mode

The cluster n-gram model has been demonstrated as an effective way to deal with the data sparseness problem which exists in the word n-gram model, but this method sacrifice some predictable ability. To solve this problem, a language model based on cluster n-grams with *n* increased selectively to trade compactness for performance has been developed. In this paper, an Absolute Weighted Difference method is presented to construct vari-gram model.

For compactness storage category n-grams, we employ a tree data structure associating each node with a particular word category. Each path represents a distinct history, such as path C_5, C_4, C_3, C_2, C_1 means that the five words before the current word correspond belong to categories C_5, C_4, C_3, C_2, C_1.

Let C represents the category which the current word belongs to; h_L represents the history (where L is the length of the history); category C_{new} is the extended history before h_L. Then we can use the Absolute Weighted Different method to measure the change that the distribution $P(\cdot \mid h_L)$ is replaced by $P(\cdot \mid C_{new}, h_L)$.

$$\Delta_{diff} = \sum_C [Num(C_{new}, h_L, C) - D(Num(C_{new}, h_L, C))] \times \left| \log P(C \mid C_{new}, h_L) - \log P(C \mid h_L) \right| \qquad (7)$$

Where $Num(C_{new}, h_L, C)$ represents the number of times the sequence C_{new}, h_L, C occurs in training corpus, and $D(Num(C_{new}, h_L, C))$ is a discount function.

The algorithm of construct the variable-length Category-based L-gram language model is as:

1. initialization : L=-1

2. L=L+1

3. Grow: Add level #L to level #(L-1) by adding all the (L+1)-grams occurring in the training set for which the L-grams already exist in the tree.

4. Prune: For every (newly created) leaf in Level #L, computer the Δ_{diff}, and discard the leaf if the value of Δ_{diff} is smaller than a predetermined threshold.

5. Termination: If there are a nonzero number of leaves remaining in level #L, go to step 2.

After getting the language tree, we can compute the perplexity of the test corpus by using the following formula:

$$perp = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log(P(w_i|h(w_i)))} \quad \textbf{(8)}$$

Where $h(w_i)$ represents the category sequence that the longest path which the category (containing word w_i) in the tree corresponds to.

5 Experimental Results

We use two annotated corpora selected from People's Daily newspaper 1998 for training and testing. Firstly, to compare the greedy clustering method based on minimum entropy and the hierarchical clustering method based on word similarity, we take a 2M corpus for testing. Secondly, we use a 5M corpus to generate the language model tree, and compare the Vari-gram clustering-based model with the models based on word trigrams and trigram clustering. The experiment results are demonstrated in table 1 and table 2.

Table 1. Clustering experiment results of two algorithms

Clustering algorithm	Greedy algorithm	Algorithm based on similarity
perplexity	283	218

From table 1, it can be seen that the perplexity is reduced from 283 to 218, and word clustering algorithm based on similarity has better performance than conventional greedy clustering method.

Table 2. Performance of three models

model	Word trigram	Trigram cluster-based	Vari-gram
perplexity	243.73	234.65	219.14

Table 2 shows that the Vari-gram language model has the highest performance among the three models.

References

1. Takuya Matsuzaki,Yusuke Miyao, An Efficient Clustering Algorithm for Class–Based Language Models[A]. Proc of the 7th Conf on Natural Language Learning at HLT-NAACL[C], 2003, 119-126.
2. Ido Dagan et al. Context word similarity and estimation from sparse data. Computer Speech and Language, 1995, 9(2): 123-152.
3. Niesler T R, Woodland P C. A variable-length category-based n-gram language model. In: Proce the International Conference of Acoustics Speech and Signal Processing, Atlanta, 1996, 164-169.
4. Firth, John Rupert. 1957. A synopsis of linguistic theory 1930-1955. In Philological Society, editor, Studies in Linguistic Analysis. Blackwell, Oxford, pages 1-32. Reprinted in Selected Papers of J. R. Firth, edited by F. Palmer. Longman, 1968.
5. Christopher D Manning, Hinrich Schutze, Foundations of Statistical Natural Language Processing. London: The MIT Press, 1999.
6. Cutting, D. R.,Karger, D. R., Perdersen, J. R, and Tukey, J. W. Scatter/garther: A cluster-based approach to browsing large document collections. In SIGIR 92.
7. Lee, Lillian. Similarity-Based approaches to Natural Language Processing. Ph.D. thesis, Harvard University, Cambridge, MA. 1997.
8. Karov, Yael and Shimon Edelman. Learning similarity-based word sense disambiguation from sparse data. In Proceedings of the Fourth Workshop on Very Large Corpora, Copenhagen, Denmark, 1996, 42-55.