LANGUAGE MODELING USING X-GRAMS

Antonio Bonafonte and José B. Mariño
{antonio,canton}@gps.tsc.upc.es

Universitat Politècnica de Catalunya
c/Gran Capità s/n
08034 Barcelona (SPAIN)

ABSTRACT

In this paper, an extension of *n-grams* is proposed. In this extension, the memory of the model (n) is not fixed *a priori*. Instead, first, large memories are accepted and afterwards, merging criteria are applied to reduce complexity and to ensure reliable estimations. The results show how the perplexity obtained with *x-grams* is smaller than that of *n-grams*. Furthermore, the complexity is smaller than *trigrams* and can become close to *bigrams*.

1. INTRODUCTION

Language modeling has been studied under two different points of view. First, as a problem of grammar inference: in this case the model has to discriminate the sentences which belong to the language from those which do not belong. Second, as a problem of probability estimation.

If the model is used to recognize (usually speech, but also printed characters or other pattern recognition taks) the decision is usually based on the *maximum a posteriori* rule. The best sentence L is chosen so that the probability of the sentence, knowing the observations O, is maximized:

$$argmax_L \{ p(L/O) \} = argmax_L \{ p(L) f_o(O/L)/ f_o(O) \}$$

The rule of the language model is to provide a good estimation of $p(L)$ for each sentence which has to be evaluated on the maximization. If a grammar is used it should be a stochastic grammar where a probability is assigned to each rule of the grammar.

If the sentence L is composed of m words, then the probability can be expressed as

$$p(L) = p(w_1 w_2 \dots w_m) = \prod_{i=1}^{m} p(w_i / w_1 \dots w_{i-1})$$

The number of parameters to estimate becomes intractable as the length of the sentence (m) increases. *N-grams* are the most extended method to reduce this number approximating the probability of a word as if only the n-1 most recent words have influence. Thus,

$$p(w_i / w_1 \dots w_{i-1}) = p(w_i / w_{i-n+1} \dots w_{i-1})$$

For each *history* $<w_{i-n+1} \dots w_{i-1}>$ a probability distribution has to be estimated. If the lexicon size is K, the number of distributions is K^{n-1}, and the number of probabilities to be estimated are K^n. Even for a lexicon of moderated size (K=1000), the number of probabilities to be estimated for *trigrams* (n=3) is larger than 1000 millions.

The training material to estimate such number of parameters is always sparse. Therefore, the *maximum likelihood* estimator is not appropriated. A number of proposals have appeared to smooth the probabilities, being *back-off* [1] the most standard reference.

The value of n is usually chosen small (2 or 3) in order to reduce the number of parameters of the model. In this way the estimation is more reliable. Furthermore, both, the storage needed and the recognition search space, are smaller.

However, the *n-gram* models can also be studied from a grammatical point of view [2] since the *n-grams* are a well known subset of the regular grammars: the *k-testable* grammars in strict sense [4]. Therefore, the model can be represented as a finite state automaton. Next section illustrates how the number of states is the same than the number of histories observed in the training data. Therefore, as far as the training corpus is finite, the size of the model does not grow exponentially with the value of n.

Furthermore, the goodness of the estimation of the probabilities can be established by other criteria which are not the length of the conditioning history. Then, the number of conditioning words depends on each particular case. That is the reason why we have name these models as *x-grams*.

The paper is organized as follows. Next section illustrates how the *n-grams* can be represented as a finite state automaton (*FSA*). Furthermore, the number of states of the *FSA* and perplexity for different values of n are evaluated. This representation allows the use of some new smoothing methods, as those proposed by Bordel et al [2]. Some methods are proposed on section 3 to reduce the perplexity of the model. Section 4 is the core of the paper and introduces two criteria which are used to validate the estimation of probabilities. The use of these criteria allows to control the tradeoff between goodness of the model and complexity. For instance, it is shown how *x-grams* can reduce significantly the complexity with respect to *trigrams* with the same performance. On the other hand, if the complexity of the model is the same than *trigrams*, the perplexity decreases.

This research was supported by CICYT under contract TIC95-0884-C04-02

2. N-GRAM REPRESENTATION

If a proper method is used to smooth *n-grams*, the number of zero probabilities is null. However, the smoothing methods usually allows a compact representation. In this section we show a proper representation of a *n-gram* which has been smoothed by the back-off [1] method. This representation allows low storage for the probabilities values and fast access. This representation will be used in the rest of the paper.

First, we briefly review the theory of back-off. The basic idea is that if a history $<w_{i-n+1} \ldots w_{i-1}>$ is not present in the training data then $p_b(w_i / w_{i-n+1} \ldots w_{i-1})$ is approximated by $p_b(w_i / w_{i-n+2} \ldots w_{i-1})$. On the other hand, if $<w_{i-n+1} \ldots w_{i-1}>$ exists, then some discount ($dC_n \leq 1$) is applied to the relative frequency so that a mass of probability is reserved. This mass can be distributed among those words which have not been observed following $<w_{i-n+1} \ldots w_{i-1}>$ on the training data. Therefore,

$$p_b(w_i / w_{i-n+1} \ldots w_{i-1}) =$$

$$\begin{array}{ll} p_b(w_i / w_{i-n+2} \ldots w_{i-1}) & \text{if } C(w_{i-n+1} \ldots w_{i-1}) = 0 \\ dC_n \dfrac{C(w_{i-n+1} \ldots w_i)}{N p_b(w_{i-n+1} \ldots w_{i-1})} & \text{if } C(w_{i-n+1} \ldots w_i) > 0 \\ R(w_{i-n+1} \ldots w_{i-1})\, p_b(w_i / w_{i-n+2} \ldots w_{i-1}) & \text{if } C(w_{i-n+1} \ldots w_i) = 0 \end{array}$$

In the above expression, $C(h)$ gives the number of times that the history h appears on the training data, R is a normalization constant chosen so that the stochastic restriction is verified and N is the total number of words (and *n-grams*) in the training data.

N-grams can be represented by *FSA* in the following way: for each possible history $<w_{i-n+1} \ldots w_{i-1}>$ a state is defined. Each transition departing from this state is labeled by a word w_i and the probability $p(w_i / w_{i-n+1} \ldots w_{i-1})$. The end of the transition is the state associated to the following history, i.e., $<w_{i-n+1} \ldots w_i>$. The probability of a sentence $p(w_1 \ldots w_m)$ can be computed multiplying the probabilities of a path of the automaton. Furthermore, the path can be determined deterministically because there is exactly one transition leaving each state for each word of the lexicon.

The number of possible histories increases exponentially with the memory $(n-1)$ of the *n-gram*. However, if a history does not exist on the training data, then the associated state is not needed. For instance, suppose that being at state $<w_{i-n+1} \ldots w_{i-1}>$, w_i arrives. The transition labeled with the pair $\{w_i,\ p(w_i / w_{i-n+1} \ldots w_{i-1})\}$ should go to state $<w_{i-n+2} \ldots w_i>$. However, if this history does not exists on the training data, then when a new word w_{i+1} arrives, the probability $p(w_{i+1} / w_{i-n+2} \ldots w_i)$ is backed off to $p(w_{i+1} / w_{i-n+3} \ldots w_i)$. Therefore, the transition labeled with w_i can go directly to $<w_{i-n+3} \ldots w_i>$. Therefore, the state $<w_{i-n+2} \ldots w_i>$ is not needed.

Figure 1 shows the number of states of a *FSA* which represents a *n-gram* as a function of *n*. The *n-gram* has been applied to model geographic inquires to a database. 14.000 inquires have been used to estimate the probabilities while 1.000 have been reserved to evaluate the perplexity on the test set. The lexicon consist of around 1400 different words. This task, referenced on [2,3], will be used on the rest of the paper.

It can be seen how the number of nodes increases significantly with the value *n*. However, the increasing is far away from being exponential. For instance, for $n = 8$, the number of different possible histories exceeds 10^{22}, but less than 150,000 have been observed on the training data.

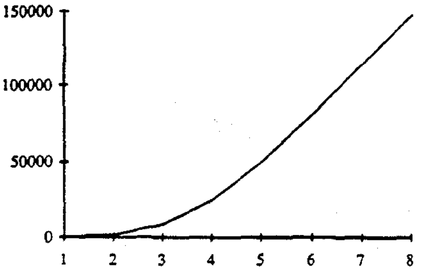

Figure 1: Number of states as a function of the memory (*n*) of the *n-grams*.

3. SYNTACTIC SMOOTHING

The back-off method reviewed in last section bases its fundaments on the join distribution probabilities: $p(w_{i-n+1} \ldots w_i)$. However, the method has some difficulties. For instance, following suggestions of Katz [1], the constants dC_n can become zero. In that case no probability is reserved for unseen events. To avoid this situation a minimum probability mass has to be reserved.

Another possibility is to smooth directly the conditioning probabilities. This can be viewed as a smoothing of the probabilities of transitions leaving each state of the *FSA*.

Bordel et al [2] proposed to reserve a mass probability Q at each state E, which is distributed among the unseen words departing this state. The estimation of Q proposed by Bordell is

$$Q = \frac{r}{q+r}$$

where r is the number of different transitions departing the state E which has been seen on the training; q is the number of times that state Q has been visited while parsing the training data. For instance, if all transitions have been used once ($r=q$), the mass probability reserved is 0.5.

The above expression estimates the reserved mass probability based on the number of different words which have been observed leaving a state. On the other hand, the original back off method estimates mass probability based on words which have been observed leaving states *exactly once*. Other expressions to estimate the reserved mass probability have been derived on [3]. The lower perplexity is obtained if Q is estimated as

$$Q = \frac{r'}{q+r'} \qquad r' = \sum_{\forall t} f(C(t))$$

t refers to the transitions leaving the state E, and $C(t)$ is the number of times that transition t is used to parse the training data. The method proposed by Bordel, equivalent to use a function $f(.)$ which always returns 1, gives better results than the original back-off method.

Still, slightly smaller perplexities are obtained if function $f(.)$ returns smaller values for higher $C(.)$. Particularly, best results are obtained if f decreases linearly, being zero for values higher than ten. Note that in this case, the mass probability reserved for unseen words is smaller. Figure 2 shows the perplexity on the test set for this smoothing method compared with original back-off. The graphs are plotted as a function of the memory n of the *n-gram*. The following points can be observed:

- The perplexity obtained with *4-grams* is noticeable smaller that with *trigrams*.
- For high values of n, the syntactic back-off is better than the original back-off.
- The minimum perplexity for syntactic back-off is obtained for *5-grams*. However, the difference with *4-grams* is small.

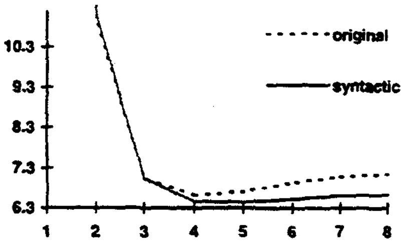

Figure 2: Perplexity as a function of the memory n of *n-grams* for original back-off smoothing and syntactic smoothing.

4. X-GRAMS: SIMPLIFYING THE FSA

As it has been stated in the introduction, the *n-grams* is the language model more extensively used in speech recognition. The value of n is chosen small, usually 2 or 3, in order to avoid computational and estimation problems. In section 2, we have shown how the size of the *n-gram* does not increases exponentially with n. On the other hand, in section 3, we have shown how the best estimation is achieved for $n = 5$. However when *trigrams* and *5-grams* are compared, it is observed that a high prize has to be paid to decrease the perplexity (*PP*):

- *trigram*: 8,631 states; *PP* = 7.03
- *5-gram*: 49,697 states; *PP* = 6.45

The problem is that if *5-grams* are used, all the words are conditioned by the 4 preceding words. However, it can happen that the probabilities estimated for these long histories are not always reliable. On the other hand, even if the estimation of the probabilities is good, perhaps it gives no additional information with respect to the probability given the three preceding words.

To cope with the problem, the idea is to apply a merging-state algorithm. Each state associated to a history $<w_{i-m}...w_i>$ is candidate to be merged with state $<w_{i-m+1}...w_i>$. Two criteria have been applied:

1. The states are merged if the number of times that history $<w_{i-m}...w_i>$ has been observed on the training data is smaller than k_{min}. A similar procedure which is usually applied to *n-gram* estimation is to force zero to those *counts* with a low value. However, in our case, the value of $C(w_{i-m}...w_i)$ is used to estimate $p(w_i/w_{i-m}...w_{i-1})$
2. The states are merged if the information of distribution $p = p(w/w_{i-m}...w_i)$ is similar to that of distribution $q = p(w/w_{i-m+1}...w_i)$. The difference between p and q is measured by the divergence, a well known function in information theory. If the size of the lexicon is J then the divergence D is defined as:

   $$D(p//q) = \sum_{j=1}^{J} p(i) \cdot log \frac{p(j)}{q(j)}$$

   The divergence is zero only if p equals q. On other cases it is greater than zero. If the logarithms are taken to the base 2, the information is measured in *bits*. If natural logarithms are taken, the measure is in *nats*.

If state $<w_{i-m}...w_i>$ is merged with state $<w_{i-m+1}...w_i>$, then the transitions which arrived to $<w_{i-m}...w_i>$ have to be addressed to $<w_{i-m+1}...w_i>$.

Tables 1 and 2 show the perplexity and the number of states when either criterion 1 or 2 are applied.

k_{min}	# states	perplexity
1	49,697	6.41
3	12,369	6.40
5	8,139	6.47
7	6,459	6.54
9	5,510	6.60

Table 1: Perplexity and number of states of the *FSA* needed to represent a *5-gram* and perplexity as a function of the minimum number of times that a history has to be observed in the training data in order to include the associated state.

The analysis of table 1 shows how the number of states can be reduced greatly without decreasing the perplexity. In fact, if k_{min} is 3, the perplexity is better than if merging is not applied.

D_{min}	# states	perplexity
0.1	18,519	6,40
0.2	12,550	6,45
0.3	9,395	6,53
0.4	7,415	6,67
0.5	5,800	6,85

Table 2: Perplexity and number of states of the *FSA* needed to represent a 5-*gram* and perplexity as a function ot the minimum discrimination of each state, measured by the divergence function in *nats*.

On the other hand, it can be observed how the perplexity is always smaller than the value obtained with *trigrams*. Furthermore, for large values of k_{min}, the number of states is much smaller than with *trigrams*.

The results of using the divergence as merging criterion are similar. However, for the same number of states the first criterion gives better performance.

The states which are merged using each criteria are not the same. An analysis reveals that there are a large number of states with low occupancy (bad estimation of the probabilities) and *other* states which present little discrimination. Therefore, both criteria can be applied simultaneously on the merging algorithm.

For instance, if the value of k_{min} is fixed, the number of states of the *n-gram* can be controlled adjusting the minimum discriminations required. For instance, table 3 illustrates this idea showing the perplexity and the number of states as a function of the discrimination when k_{min} is fixed to 3.

D_{min}	# states	perplexity
0.0	12,369	6.40
0.1	5,977	6.41
0.15	5,156	6.41
0.2	4,566	6.46
0.25	4,172	6.54
0.3	3,841	6.56

Table 3: Perplexity and number of states of the *FSA* needed to represent a 5-*gram* and perplexity as a function of the minimum discrimination of each state, measured by the divergence function in *nats*. Each state should be visited at least 3 times.

It is shown how it is worth to combine both criteria. For instance, perplexity smaller than 6.5 can be obtained with around 4,500 states. However, if only one criterion is applied, either 8000 or 12000 states are needed.

In fact, as changes on the perplexity values are small, the conclusion which can be drawn from table 3 is that complexity of the *n-gram* can be enormously reduced without degrading significantly the performance of the language model.

The experiments reveal how the criteria to merge states can effectively select the states which are relevant from those which are not. The idea beyond *x-grams* is that, instead of approximating the value of $p(w_i/w_1 \ldots w_{i-1})$ by $p(w_i/w_{i-n+1} \ldots w_{i-1})$, it is better to accept all the states associated to histories of all the lengths as candidates and, afterwards, select those which are *relevant*. In practice, this can be implemented by merging states from an *n-gram* with sufficient large value of *n*.

Table 4 shows the number of states and the perplexity of x-*grams* for two cases. In the first case (*x-gram*(1)), the parameters of the merging criteria are $k_{min} = 5$ and $D_{min} = 0.35$ nats. These parameters are chosen so that the number of states is low. The second case (*x-gram*(2)) uses $k_{min} = 2$ and $D_{min} = 0.1$ nats. In this case, the parameters are chosen to have low perplexity. It should be noted that the choice of the parameters is not a critical issue; both criteria exhibit a smooth evolution. To ease comparisons, the performance of *bigrams*, and *trigrams* is included on the same table. It can be shown how *x-grams* outperform *n-grams* with controlled complexity. As a curiosity, the maximum length of histories associated to the states of *x-grams* (1) and (2) happens to be 7 and 8 respectively.

Model	# states	perplexity
bigram	1,391	11.00
trigram	8,631	7.03
x-gram(1)	2,943	6.67
x-gram(2)	7,661	6.39

Table 4: Perplexity and number of states of the *FSA* needed to represent *x-grams* compared with *bigrams* and *trigrams*. *x-gram*(1): $k_{min} = 5$, $D_{min} = 0.35$ nats; *x-gram*(2): $k_{min} = 2$, $D_{min} = 0.1$ nats

5. DISCUSSION

In this paper *x-grams* has been proposed as an extension of *n-grams*. The evaluation over a task with low perplexity and moderate lexicon (1300 words) shows the good performance of *x-grams*. The goodness of this model for more complex tasks will be evaluated on the near future.

6. REFERENCES

1. S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", *IEEE Trans. on ASSP*, Vol. ASSP-35, N° 3, pp. 400-401, March 1987.

2. G. Bordel, I. Torres and E. Vidal, "Back-off smoothing in a syntactic approach to language modelling", *Proc. of the ICSLP'94*, pp. 851-854, Yokohama, 1994.

3. A. Bonafonte, *Speech Understanding on Semantic Restricted Tasks*, PhD. Dissertation, Universitat Politècnica de Catalunya, Barcelona, 1995.

4. E. Segarra, *Una Aproximación Inductiva a la Comprensión del Discurso Continuo*, PhD. Dissertation, Universidad Politécnica de Valencia, 1993.