On structuring probabilistic dependences in stochastic language modelling

Hermann Ney Ute Essen Reinhard Kneser

1994

On structuring probabilistic dependences in stochastic language modelling
Hermann Ney, Ute Essen and Reinhard Kneser
Philips GmbH Forschungslaboratorien, Aachen, P.O. Box 1980, D-5202] Aachen, Germany
Abstract

In this paper, we study the problem of stochastic language modeiling from the viewpoint of introducing suitable structures into the conditional probability distributions. The task of these distributions is to predict the probability of a new word by looking at M or even all predecessor words. The conventional approach is to limit M to 1 or 2 and to interpolate the resulting bigram and trigram models with a unigram model in a linear fashion. However, there are many other structures that can be used to model the probabilistic dependences between the predecessor word and the word to be predicted. The structures considered in this paper are: nonlinear interpolation as an alternative to linear interpolation; equivalence classes for word histories and single words; cache memory and word associations. For the optimal estimation of nonlinear and linear interpolation parameters, the leaving-one-out method is systematically used. For the determination of word equivalence classes in a bigram model, an automatic clustering procedure has been adapted. To capture long-distance dependences, we consider various models for word-by-word dependences; the cache model may be viewed as a special type of self-association. Experimental results are presented for two text databases, a Germany database and an English database.

Introduction

The need for a stochastic language model in speech recognition arises from Bayes' decision rule for minimum error rate (Bahl, Jelinek \ Mercer, 1983). The word sequence W,...Wy to be recognized from the sequence of acoustic observations x,...x7 is determined as that word sequence w,...w, for which the posterior probability Pr(w,...Wy|X,...X7) attains its maximum. This rule can be rewritten in the form:

arg max \Pr(w,.. Wy... Wy

a)

where Pr(x,...x,|W,...wy) is the conditional probability of, given the word sequence w,...Wy, observing the sequence of acoustic measurements x,...x, and where Pr(w,...Wy) is the prior probability of producing the word sequence w,...wy.

The task of the stochastic language model is to provide estimates of these prior

probabilities Pr(w,...w,). Using the definition of conditional probabilities, we obtain the decomposition:

N

Pr oy) T] Prbvryf yey @) n=

For large vocabulary speech recognition, these conditional probabilities are typically used in the following way (Bahl e: a/., 1983). The dependence of the conditional probability of observing a word w, at a position n

is modelled as being restricted to its immediate M predecessor words w,\..W, ¯, −Theresultingmodelisthatof aMarkovchaina
1)−grammodel.ForM = 1andM = 2, weobtainthewidelyusedbigramandtrigrammodels, respectively.Thesebigramandtrigr
gramapproachesaresmoothedwithmoregeneraldistributionsbylinearlyinterpolatingtheprobabilitiesbetweenthedistribution
gramapproachandthelinearinterpolationcanbeviewedaschoosingaspecialtypeof functionaldependenceforthe conditionalpro
−W, ¯;).Sothequestionariseshowweshouldchoosethetypeofthefunctionaldependencebetweenthepreceedingwords Wa−
me − − − ˇě, \, andthewordw, underconsideration.

In this paper, we study various approaches towards introducing some type of structures into these conditional probabilities. Some of these structures fit into the M- gram model and allow refinements, others are based on completely different concepts and are aimed at modelling long distance dependences in texts. From a general point of view, the problem is to find suitable structures for modelling the probabilistic dependences for words in natural language. In this light, the following questions and approaches will be considered.

Smoothing methods, The linear interpolation between specific and more general distri- butions specifies a certain type of functional dependence, What other types of smoothing can be thought of? We will introduce a nonlinear scheme and contrast it with linear interpolation.

Simulating unseen events. The interpolation scheme, be it linear or nonlinear, requires some interpolation parameters to be suitably chosen. The smoothing must account for events that were not seen during the training of a specific distribution. This effect of 'unseen events can be efficiently modelled by the so-called leaving-one-out method which is a special type of cross-validation. In combination with maximum likelihood estima- tion, this method will be shown to provide an efficient method for finding the unknown interpolation parameters.

Equivalence classes for word histories and word categories. Linguistically defined word categories, sometimes referred to as parts of speech, are sometimes useful for increasing the generalization capabilities of a language model by combining them with M-gram word models. We wili present a technique for automatically finding word categories Probabilistic dependences in stochastic language modelling 3

using statistical principles. This technique is basically a statistical clustering procedure that works iteratively in the spirit of decision-directed learning.

Word associations. There are long distance dependences in texts, where the distance between words may go up to a thousand and more words. The occurrence frequency of a word, in particular that of a less frequent word, docs depend on the topic of the text and other words used in the same text passage. A framework of word associations will be presented to approximately handle these effects so that the language model automatically adapts to topic changes in the text passages, In this framework, the depen- dence between a word and its preceding word sequence is decomposed into pairwise dependences between the word and each of its predecessor words. The cache model that has been recently introduced by Kuhn and de Mori (1990) can be viewed as a special self- associative model in this framework.

Strictly speaking, to evaluate the quality of a stochastic language model, we would have to run a whole recognition experiment. However, in a first approximation, we can separate the two types of probability distributions in Bayes' decision rule and confine ourselves to the probability that the language model produces for a sequence of (test or training) words w„ n=1,...,N. To normalize this prior probability with respect to the number NW of words, we take the N-th root and take the inverse to obtain the so-called corpus (or test set) perplexity (Bahl ef a/., 1983):

PP:=[Pr(w,...wy) Q)

Inserting the decomposition into conditional probabilities of Equation (2) and taking the logarithm, we obtain:

N

© log Pr(w, |)... Wy =1)- (4)

nel

log PP=

Throughout this paper, we will use the term 'corpus perplexity' rather than 'test set perplexity', because we will consider perplexities of both training and test data. The above equations show that the corpus perplexity is the geometric average of the reciprocal probability over all N words. Apart from the constant (—1/N), the corpus perplexity is identical to the probability or likelihood. Therefore minimizing the corpus perplexity is the same as maximizing the log-likelihood function.

There is a remarkable implication in the definition of corpus perplexity. If a word in the corpus is assigned a probability of zero by the language model, the perplexity will be infinitely large. This is one of the real challenges for the language model: the prediction of the next word should be as good as possible without excluding any of the words of the vocabulary. Thus it can be seen that the definition of the corpus perplexity includes the coverage problem. However, there are some shortcomings in the definition of the (corpus) perplexity. The perplexity is merely a single averaged scalar-valued quantity; there is no information about the local variations across the corpus. It would be straightforward to define a variance; an even more informative method would use a histogram over local probabilities, ic. reciprocal perplexities. Another potential drawback is the fact that neither the interaction with the acoustic probabilities Pr\x,...X7|W,...W„) nor the effect on the search effort are taken into consideration.

N. Ney et al.

Experimental tesis will be presented for stochastic language modelling on a German text corpus of 100000 words and on an English text corpus of 1-1] million words. However, it should be stressed that the techniques that will be presented are by no means limited to language modelling. They may be useful in similar areas, where probabilistic dependences are modelled and the number of free parameters to be trained is large. Examples are machine translation, information retrieval and database queries, expert systems and diagnosis systems, machine learning, and any other field in which information is processed, decisions are taken and the complexity of the dependences require some suitable selection of the functional structures.

The organization of the paper is as follows. In the next section, we present a rather general view of the smoothing and interpolation problem including nonlinear tech- niques. Then the use of the leaving-one-out method is developed for the estimation of the interpolation parameters. In Section 4, we apply the concept of equivalence states and word categories to formulate structural dependences and reduce the number of free parameters in word-based bigram models. Section 5 introduces the concept of word associations to cover long-distance dependences between words; the cache model can be interpreted as a self-association. Finally experimental results are presented to test these probabilistic structures.

Smoothing methods

Multinomial distribution and maximum likelihood estimation

Any type of language model faces the problem that the amount of training data is limited and always too small to observe the typical events often enough. Here, we will use the term event or event class to describe the type of observations we are considering. When we are looking at word bigrams, the events will be the possible word pairs. In discussing conditional bigram probabilities for a fixed predecessor word, the events will be the single words that may follow the fixed predecessor word.

Let us denote the event classes under consideration by k=1,...,K, their sample counts by Mk), i.e. the frequency how often class k was observed in the training text, and the corresponding probabilities by p(x), i.e. the chance of success of class K. For a set of N observations (in training or testing), we have N independent trials with K possible outcomes, where the sample counts N(x) denote the number of trials resulting in class or outcome k, The distribution of the sample counts N(k) is referred to as multinomial (or polynomial) distribution (Lehmann, 1983):

PAN, 5 MED) =e TT ote ©) p= NO with the following constraints:

K K

YNWQEN and ¥ pk)=1. 6)

kel k=l Probabilistic dependences in stochastic language modelling 5

As we will see later, it is helpful to partition the event classes k into equivalence classes according to their sample counts r= N(k) (Good, 1953), which is referred to as symmetry requirement (Nadas, 1985). Let 1, be the number of event classes which occurred in the training text exactly r times and p,> 0 be the corresponding probability, i.e. we define:

Pry = PR).)

The above constraints can then be rewritten as:

R R

Vim,=N and ¥ pn,=1, (8)

rd r=0

where we have: n,= 0 for r> R. Note that the sum over p, has a non-zero contribution for the index r=0 due to the requirement p,> 0. It is instructive to compare the formulation of the log-likelihood function F in terms of the sample counts N(k) and of the equivalence class counts n,:

F= y N(k) log p(k) and F= Sin, log p,. (9)

kel r=1

Maximum likelihood estimation (Lehmann, 1983) for the underlying multinomial distribution results in the relative frequencies as probability estimates:

Nk)

N

p(k)= and p,=

N' (10)

As an example of the smoothing problem, consider bigram modelling for a vocabulary of W= 20000 words: A training text of N= 10 million words can cover at most 2°5\the K=W*=400 million possible word pairs. The sparseness of data in language modelling is then characterized by the following inequalities:

n<Nan<K. ay

In addition to these strict inequalities, the experimental results show the following typical property of the n, sequence in the case of word bigrams (or trigrams):

3n,<2n, <n. (42)

In maximum likelihood estimation, a probability of zero is assigned to all unseen events, which in a typical situation make up to 95\ing to these rare but nevertheless possible events would have no chance to be hypothesized or even recognized in the speech recognition process.

N, Ney et al.

Floor method and linear interpolation

To guarantee non-zero probability estimates, we can modify the sample counts N(k) by adding a floor value which is chosen proportional to probabilities g(k) of a less specific distribution, e.g. a unigram distribution or a uniform distribution. After renormaliza- tion, we then obtain the probability estimates:

1) =O (3)

It is interesting to note that a strictly Bayesian estimation using a uniform prior distribution and a squared error as cost function leads to adding an offset of 1 to all counts N(k) and renormalization, which is usually referred to as Jeffrey's estimate (Lehmann, 1983). However, the experimental results show that this type of Bayesian approach (with the choice = K) overestimates the probabilities of events with small sample counts r. Defining a new parameter 2

=! wy A= app with 0<A<]1, (4)

we obtain the usual interpolation formula (Jelinek, 1985):

pth)= (1-2 XO+ aatty as)

This equation defines what will be called the linear interpolation model with interpola- tion parameter 4. The characteristic property of the linear interpolation model is that a weighted average of specific and more general distributions is computed and that the weights are independent of the sample counts N(x). In other

words, each sample count N(k) is reduced ("discounted") by a value AN(k). However, it can be argued that the higher the sample counts N(«), the higher their contribution should be in the interpola- tion model. A possible solution is to introduce bins for the counts V(x) as described in Jelinek \ Mercer (1980).

Discounting models and nonlinear interpolation

Another approach is to consider more general discounting methods in the spirit of Katz's technique (Katz, 1987). We introduce a general discounting function d:k—d(k) that has to be subtracted from every sample count N(x) and obtain the following smoothing formula:

Na a(k)

N

pA)= + Oldla(k), (16)

where the discounted probability mass Q[d] depends on the discounting function das a whole and is defined as: Probabilistic dependences in stochastic language modelling 7

K

Qld\=— ¥ dk). (7) ket

We now assume a special model where the sample counts N(k)>0 are discounted by a constant value D with

0<D<1, (18)

and then the discounted probability mass (K—n,)D/N is redistributed over all events k=1,...,K according to the distribution g(k):

r= maxlN)— DO, Kk D mu (k). (19)

Aconstant value D with the range constraint 0< D<1 can be intuitively interpreted as a correction that lies well within the range of the "discretization noise" of the discrete counts N(k). This nonlinear interpolation model has the following properties:

—The weight of the more general distribution g(k) is proportional to (K—1,), which is the total number of different events \ that were observed. This property is particularly attractive for modelling conditional probabilities: if a given word predecessor is followed by only one or a few different words, the smoothing effect will be much smaller than in the case where it is followed by many different words.

—Setting D=1 amounts to pooling all singletons, i.e. events k with N(k)= 1, with the unseen events. As Katz (Katz, 1987) pointed out in a similar context, there is not much difference between seeing an event just once or not at all.

—Although the interpolation is nonlinear, there is always a smoothing effect in that a weighted average between the two distributions is computed. This is different from Katz's backing-off approach (Katz, 1987), where a choice must be made between the more specific and the more general distribution.

It is straightforward to generalize this discounting model along different directions. At the level of conditional bigram probabilities p(w|v), we have, assuming N(v)>0:

rv) =A S009) + Oat), 20)

where in the general case, the gained probability mass Q,[¢] depends now on the predecessor word v and is computed as:

Qld FesPae. Ww). (21)

N. Ney etal.

By this definition of Q,\d\, it is ensured that the conditional probabilities p(w\v\ sum up to unity.

Now for the model of absolute discounting, we select a constant value D,>0 for each depressor word v and define the discounting function:

d(v.w)= min[D„N(v,w)] (22)

The dependence of Q,[d] on the model d(v,w) is reduced to the parameters D, only, and the notation will be Q,(D,). The conditional probability can then be rewritten as:

Hon[yy= MEIN PA. 9.(D.)a(w) 23)

Assuming the parameters D, in the range [0,2], we have:

Wi) + DIW— W0)- W0)]

Wo) if 1<D,<2

Od) = (24)

Diw-W)]

NO) if0<D,<1

where W is the vocabulary size, W,(v) is the number of words that never followed v and W,(7) is the number of words that followed v exactly once.

Simulating unseen events: leaving-one-out

The basic principle

The unknown parameters in the discounting models can be automatically determined with the so-called leaving-one-out method. This method can be obtained as an extension of the held-out method (Duda \ Hart, 1973), where the training text is split up in two parts: a "retained" part for estimating the relative frequencies and a "held-out" part for estimating the optimal interpolation parameters. Here, the training text is divided into N-1 samples as the "retained" part and only 1 sample as the "held-out" part; this process is repeated N times so that all N partitions with | "held-out" sample are considered. The basic advantage of this approach is that all samples are used both in the "trained" part and in the "held-out" part and thus a very efficient exploitation of the training text is achieved,

The formal description of the method is as follows. Using the symmetry principle as explained above, the probability estimates depend on both the counts N(k) and the general distribution q(x). To express these dependences, we use the notation:

PA)= PIN); 4(k)]- (25) Probabilistic dependences in stochastic language modelling 9

The training corpus consists of a sequence of observations k„n=1,...,N. The log- likelihood function of the leaving-one-out formalism can then be written as:

N K

F= ¥) log plN(,)— a@]= Y NG) log pV) 1:4(4)]- (26) Ke

n=l =L

Tn the following, we will consider the leaving-one-out concept for the two models of linear and absolute discounting in detail.

Linear discounting

In linear interpolation, a weighted average between the relative frequencies N(k)/N and the general distribution g(k) is computed:

ain satt) = 0-98 O+ aoe) with 0<1<1, (27)

The log-likelihood function to be maximized is:!

N(k)\textasciitilde1

K

F@= YN @ log [0-aS— + Ag(\). (28) ie

1

Assuming a maximum in the range 0<4< 1, we take the partial derivative with respect to A, set it to zero and multiply the equation with (1-4):

K Kk) ¥ no [1-2] =o. 09) zn a\textasciitildeaX@at ad

Multiplying this equation by 4 and separating the term 4, we obtain:

K (k) +5 yg@g——\$_4$\\$_{\(30) Nyea^{-} OA+salty}$

' Strictly speaking, in leaving-one-out, the denominator N has to be adapted; for simplicity, we neglect this renormalization.

N. Ney et al.

=z Bl “Wy, K)

ah ¥ Nj 0 \$_{\B D}$

Mks

exit [= pot + igh)

Thus, we obtain the final result:

(32)

I

4a 44— MA) N * Na (t- — hi:

+ glk)

In the Baum formalism (Jelinek \ Mercer, 1980), this equation can be used to iteratively compute 2, Starting with 4 =0, we would get:

woe

=H (33)

G4)

1 N Lr Mk

Jen (R)> 1

m +n

A further refinement of the approximate solution is presented in the appendix.

It is instructive to note that by a special choice of q(x) the case of Katz's backing-off (Katz, 1987) is included, namely by the choice g(k)=0 for N(k)> 0. For Katz's backing off, we have, assuming an arbitrary distribution g(k):

a -0 if N(K)>0 PIN (k);q(k) = G5) ™ if N()=0

where g(k) is obtained from q(k) by suitable normalization. From Equation (32), we immediately obtain the exact solution:

ant. (36) Probabilistic dependences in stochastic language modelling ot

Absolute discounting

For the case of absolute discounting, the smoothed probabilities can be written in the following general form:

PIN (K)sq(A)] = max [oAe-?] + DE Math) with 0<D<1, G7)

Now this general form is awkward to handle. We have either to approximate the dependence on g(k) by the dependence on its average 1/K and define

b:= Dal K<1 (38)

or limit ourselves directly to the case of backing-off. In both cases, we have then for PLN (k):q(k)\:

Nik-b . "Si itnw>e PLN (K);q(K)] = (G9) bak) if Nk)=0

with 0<b<I, (40)

where the original general distribution q(k) has been renormalized to 4(k) so that the distribution p(k) sums up to unity. The log-likelihood function using leaving-one-out is:

F)= YY log (6Gk)]+ ENE) log PING) —1:q@)]. 41) k= 1 kN®>1 =const (b)+n, log 6+ ¥. mn, log (r— 1-5). (42)

rt

To find a solution with 0< b< 1, we set the partial derivative with respect to 5 to zero and separate the term for r=2 from the sum:

ny 2m, $\backslash_r h, mhye436b1 - bSyr - 1 - b$(43)

N. Ney et al.

For practically interesting cases, it is possible to derive an approximate solution in the following way. Typically, the sequence of experimentally observed values r-„ r= 1,..., R forms a decreasing sequence. Thus to a first approximation, it suffices to consider only the largest ,-dependent terms in the sum. To this purpose, we rewrite the equation for b:

ny —B(\Sy to ln-b

Since the right-hand side of the above equation is positive for all 6 with 0<6<1, we have an upper bound 6, for b:

a

b<bo= om,

<i. (45)

Here, we have assumed that n,>0 and n,>0, which is satisfied in all cases of practical interest. A more refined approximation is presented in the appendix.

Equivalence states and word categories

In this section, we will consider structures that allow us to define similarities between word pairs or within whole groups of words. To this purpose, we first introduce a notation which is better suited to the following derivations. Rather than using the word position index n, we will use the counts N(v„...¥)) of any (short) word sequence ¥„...¥,. From the set of these counts, we can recursively calculate the counts for shorter word sequences using the equation:

Nm — 1\

om

and obtain thus all counts N(vy\,...ĕ9)$s - . - N(vp)Thelog - likelihoodfunctioncanberewritten$:

N

F= ¥) log Pry] Wy — ap Mn—1) an

nat

= Yo N(y.–¥\

(4.0)

For notational convenience, we use the following convention: The lower-case symbol Probabilistic dependences in stochastic language modelling 13

P| .) is used to denote a concrete model of a probability distribution, while the symbol Pr(.|.) stands for a most general probability distributions with no or nearly no constraints. For example, in the above equation, the only constraint on Pr(v| vy...) is that its conditioning events are limited to the M preceding words.

Pairwise similarities between words

First, we consider similarities between word pairs. By similarity, we denote the property that a certain word might be replaced by some other word in a given context, e.g. a bigram or trigram context. An obvious way of defining word similarities is to find words that are likely to be predicted by the language model in the same context. Along these lines, word synonyms can be introduced as follows. We fix a word and compute what is the total log-probability with which another word would be predicted in the same context. As an example, we consider a bigram language model with conditional probabilities p(v,|v,). For a bigram, the full context of a word y, is the set of all word triples (v,v,v,). For the difference in the log-likelihood when the word y, is replaced by w in all contexts (»,¥,¥,), We have:

= DN (yy\

AF(w; . (winks plor]w p01)

Similarly, for the case that the word y, is replaced by nothing, i.e. the 'empty' word ¢, we have:

AFten= E Noon) log a aot 7" Yo) (50)

Finally, we can consider the case that the word w is inserted into a context (v,.¥)):

AF(w;2):= Y.N(v4¥p) log —Ptva| yo) (51)

v0 (vp))pC| yo)"

For a given word ¥„ the word w that produces the smallest decrease in the log-likelihood function might be called a synonym of word y,. To make this relationship symmetric, we exchange the arguments w and », in f(w;v,) and define the synonym o(v,) of a word v, by the equation:

o(v,): = arg minf\AF(w;y,) + AF(v,;)]\. (52)

wey

N. Ney et al.

Ina similar spirit, pairwise word similarities have been used in co-occurrence smoothing (Essen \ Steinbiss, 1992) and in vocabulary adaptation (Jelinek, Mercer \ Roukos, 1990).

Modelling equivalence states and word categories

Word categories or parts of speech (POS) can be viewed as an attempt to cope with the problem of sparse data in language modelling (Deroualt \ Merialdo, 1986). Typically, these word categories are based on syntactic-semantic concepts and are defined by linguistic experts. Generalizing the concept of word similarities, we can also define word categories by using a statistical criterion, which is in most cases, but does not have to be, maximum likelihood. A similar method has been described by Jelinek (1991). The approach presented in this paper will be different in a number of ways. In particular, we always make the distinction between the likelihood criterion with and without plugged- in maximum likelihood estimates. The advantage then is that we are not restricted to using the classical relative frequencies as estimates. In the section on the leaving-one-out method, we have seen that the relative frequencies are not always good estimates for unknown probabilities.

There are two levels at which we will introduce equivalence concepts: the level of word histories and the level of single words. The equivalence classes of the word histories ¥y-+.¥, for a word yy will be called states and denoted by a so-called

state mapping S : (vy... v,)+8=S(v,...¥,)- For single words, we will use the so-called category mapping G : ¥»>G(\

In the following, it is convenient to introduce special counts that are derived from the full set of counts N(v„...¥)). For a given mapping S of word histories v,...v„ we define the count N(s,v):

N(s,¥9): = x N(vy—¥\
va eS (a Ss
and another count for the states
M(s):=¥N(s,¥9)- (54)
For a given mapping G of word categories, we define the counts N(v„...„g) and N(g): Probabilistic dependences in stochastic language modelling 15
NOy eh LNG ¥) (55) va:G (v9) = 8
N@r= FY NO). (56) vo:G (v0) = \
For the two given mappings G and S, we define the count N(s,g):
N(sg):= Yo N(s,v9)- (57)
vorG (va) = £

States: equivalence classes of word histories

Using the mapping S of word histories, we have the probability model:
Pr vl Yas ++ ¥)= PCY | Sag -¥4))> (58)
The log-likelihood function can be written as:
F(S)= Nig) ¥9) log p(vg| Sy — 4) = FE M(s,¥9) log plvo|s). (59)
(ym. .Ftv0) (s.va)
Plugging in the relative frequencies of maximum likelihood estimation as estimates for (|):

Ng,

ot Fi) (60) we obtain: F(S)= ¥ N(x) log ae. 61)
(s,y0)
This criterion has the form of an entropy, which also arises in the context of hierarchical equivalence classes and CART (Bahl, Brown, de Souza \ Mercer, 1989), where a tree Structure is imposed on the mapping S. Here, no special structure for S is assumed. By using equivalence states, we can reduce the number of free parameters: |s|w— 1) in

comparison with W(W-—1) for a bigram model and *(W— 1) for a trigram model, where |S| is the number of equivalence states. However, these numbers are somewhat artificial, because even for large text databases, the number of really independent parameters is much smaller.

Word categories as membership model

Another type of structure is the membership or category model, where a word is selected from a given category. We describe the model by the inverse process: for each word yp, we define its category by the category mapping: y,>G().
We will use an index 0 in a probability distribution to distinguish a membership probability from the probability of a sequence:
Po(\word, the word » is in category g;
P(\words v„...¥„ the word y, follows.
The language model probability is then: Pr(vg| Yq. –\Using the same techniques as before, we obtain the equation for the log-likelihood function, now depending on the mapping G: FG)= YL NGy—¥\
Ome ra)
Now we insert:
$\backslash\backslash_{p}hn)PaliC{=}TGC)(64)PAGO)ry9){=}e(65)andobtain{:}FE){=}TNO\backslash$
Plugging in the maximum likelihood estimates for an ‘event’ e in the form Probabilistic dependences in stochastic language modelling 17

=X with e=g or (Vy...¥4) OF (gg Pp8)s (67) we have: F(G)=YN(q) log N(vg)+ YN vg) log NO 78) 8) vo Owe ong) Ny ¥)N(B)

Like equivalence states, membership models also lead to a reduction in the number of free parameters. The reduction results from the fact that for each conditioning event Vyg-+-Vp there are only |G|—1 free parameters rather than W— 1 free parameters in the unconstrained case, where |¢| | is the number of word categories. There is a small increase due to the membership distributions p,(.|.): their specification requires w-Ic| para- meters and W indices for the word category mapping w>G(w).

Two-sided model: states and categories

So far, we have considered one-sided models. Obviously, the two types of equivalence classes can be combined into a two-sided model with states s and categories g:

Pr vol ¥y–

)= Palo] G(\

A similar derivation as before results in:

(sg)

F(SIG)=EM) low mo)+ YMG) log (70) Using the plugged-in maximum likelihood estimates, we have: F(S,G)=Y.N(\ao NON)

Two-sided symmetric model

A two-sided symmetric model can be defined by using the word categories both for the current word and its predecessor words. In this context, it is helpful to change notations: the position will be denoted by v,v,v) and the corresponding word categories by 2,218. The mapping G is chosen to be identical for the three positions v,v,v,. For a bigram model, we have then:

Pr(vg| ,) = Pal\

N. Ney et al.

The log-likelihood function for such a symmetric model is easily derived using the equations of the preceding paragraph:

N(B818)

F,(G)= TNO) log Nooo)+ © Neues) log N@o¥ay” (73)

where N(g,g,) is defined in the usual way. Such a symmetric model leads to a drastic reduction in the number of free parameters: (\G|-1)|G| probabilities for the table Pied: W —|G| probabilities for the table MulGony and W indices for the mapping w+G(w). For a model with |G|=100 categories and W=10000 words, we have about a total of 20000 probabilities and

000 indices as compared to 10.000? probabilities for a bigram model.

Tn a similar way, we have for the symmetric trigram model:

N (828180)

F,\G)= YN) log N(vy) + Ni log ——228180/ 714 \G) py (\

Probabilistic mappings for states and word categories

So far we have considered deterministic mappings, i.e. for each word, there was exactly one word category, and for each sequence of predecessor words, there was exactly one equivalence state. Both mappings can also be stochastic: a word might be assigned to several word categories with different probabilities so that we have to use a conditional distribution p,(v.|g), i.e. the probability that given a category g, a word y, belongs to it. Similarly we have a distribution over the states q,(s|v,...v,) that is conditioned on the predecessor sequence v,.. A fully stochastic model of states and word categories predicts a word v, from the given predecessor words v„...v, in three steps: 1. Given the predecessor word sequence Vyoe++y¥„ an equivalence state s is selected using the probabilities of a membership model (S| ¥y4—¥,)- 2. Given the equivalence state s, a category g for the possible successors words is selected using the sequence probability model p(g|s). 3. Given the category g,a word v, is selected using a (second) membership model PolYo| |g). Thus according to the Bayes' theorem for conditional probabilities, we have to sum over the probabilities of all possible realizations:

Prt) = Zeal | Del dash w-—r0] oo)

From this probabilistic model, we can recover the deterministic model in the following way. For deterministic mappings, all these probabilities are zero apart from the value specified by the mapping. In particular, the probability aels| ¥y–.¥,) is 1 for the assigned Probabilistic dependences in stochastic language modelling 19

equivalence state and 0 for all other states. Hence it is clear that in the sums of the above equation, there is only one non-zero term, and we again obtain the equations for deterministic mappings.

The above equation is of importance for linguistic categories, i.e. parts of speech (POS). In the linguistically tagged LOB corpus (cf. section on experimental results), we observed that for about a quarter of the words, there was more than one POS, and in such a case, summing over all realizations is important. Without summing up, the corpus perplexity was typically 20\

3. Clustering algorithm for word categories

So far, the assumption has been that the mappings for the word categories or the equivalence states are known. Now we describe a procedure by which such mappings can be determined automatically. This automatic procedure will be developed for the two- sided symmetric model of Equation (72).

The task is to find a mapping G:w->g= G(w) that assigns each word to one of |Gj different word categories. Since these categories are found by a statistical clustering procedure, they are also referred to as clusters. The perplexity on the training data is used as optimization criterion; in other words, the optimization criterion is given by Equation (73). In the spirit of decision-directed learning (Duda \ Hart, 1973; p. 240) the basic concept of the algorithm is to improve the value of the optimization criterion by making local optimizations, which means moving a word from one category to another in order to improve the log-likelihood criterion. Thus we obtain the following algorithm:

Start with some initial mapping G:w>G(w)

Iterate until some convergence criterion is met

Loop over all words w

Loop over all clusters g'

Check how log-probability changes if w is moved from cluster g= G(w) to g'

Move word w to the cluster g' with the highest log-probability

From this description, it can be seen that the log-probability increases (or stays constant) in every iteration step. Since the probability has an upper bound of 1, the thus obtained monotone sequence is bound to converge. However, the resulting solution is only locally optimal since it depends on the start conditions.

During each iteration, moving a word w from its present cluster g=G(w) to a new cluster g' affects only those counts N(g,g,) where ¢= G(w) or g' appear as arguments. Therefore we can get the new value of the log-probability from the old one by adjusting only the corresponding counts in Equation (73). The complexity of this algorithm is

hence of the order | G)>- W- 1, where, as usually, W is the vocabulary size, i.e. the number of words to be clustered, |G| is the number of clusters and J the number of iterations. Since moving of a word to another cluster affects the clustering of the subsequent words in thé training corpus, the order in which words are moved is crucial. We know most about the frequent words and should cluster them first. Hence we sorted the words in descending order of frequency so that in each iteration the most frequent words were processed first.

The algorithm requires some initial mapping or partition G to start with. The experimental results indicate that this initial partition can be efficiently obtained in the following way. The most frequent |G|— 1 words are selected and each of them defines its own cluster. The remaining words are all put into one cluster. Other initialization schemes were found to work as well and not to affect much the final result; however, their speed of convergence may be much slower. Words of the vocabulary that do not occur in the training data do not affect the log-probability criterion of Equation (73). These words are not moved in the clustering procedure and remain in the cluster as specified by the initial partition. It is known (Katz, 1987) that the words which occur only rarely in the training text give a good estimate also for the words which do not occur at all, So the unseen words are pooled with words whose occurrence counts are sufficiently small, say 3 or 4. The final modification now is to exclude these infrequent words from the clustering procedure and keep them in the cluster "infrequent words".

The algorithm assumes a fixed value |G| of the number of clusters to be specified beforehand. So the algorithm does not give a direct way to find the best number of clusters. One way to solve this question is to use cross-validation, i.e. to divide the training corpus in two parts and then use the first part to find the clusters and optimize the number of clusters on the second part.

Word associations

Model of pairwise dependences

The models considered so far are able to cover only short distance dependences between words in a text. However, it is known that there are long distance dependences that are caused by the semantic context: depending on the topic of a text passage, certain words are more likely to occur than others. The M-gram models are difficult to use for such long distance dependences because the dependences might span across hundreds of words and more, i.e. 44= 100, and M-gram models are already impractical for M>2 or

To avoid these problems, we assume that the dependences can be decomposed into

pairwise interactions between a word w, at position n and a word w,\,, $atposition(a-m)$:

Pry | Wy ye hn = ¥ C08 | My — Ds (76)

m=

where the coefficients ¢„ and the function values ax(w|) must satisfy the stochastic constraints: Probabilistic dependences in stochastic language modelling 21

M

20 and Y (77)

m=

a(w|y)<0 and Yeo] y= I for each v. (78)

These constraints guarantee that the probabilities of the model sum up to unity. The values of a(w,| w,\,,)$donotdependonthepositiondistancembetweenthewordsw, andW, -mTheywillberegardedasprobabilitiesforwordassoci cientsc,, aSmixtureweights.Thesecoefficientsc,, defineaword-distancedependentweightoftheinfluenceofeachpredecess$

It should be noted here that the above functional form of pairwise interactions is by no means the most general one or the only possible. A more general model would be:

Ln OW tt)

Lady usw, i!)

Pr(w,|W,—y4—Wya = (79)

where the g(.;.,.) are arbitrary, but nonnegative functions that depend on both the predecessor word w,\,, $and the position distance m between the current word and the predecessor word. However, this model is probably too general to b$

max[9(W,5¥,— mi?) Yo) = Smaxfu, ol):

Prw, \Wy seer (80)

Still another functional structure is obtained by decomposing the conditional probabil- ity into a product of pairwise interactions:

Pr(Wy ap 1) $\backslash_P OW,) Pr(Wy^{-} ages Wyo[WdPriw, geWy - y) Pr(Wyage Wy)$ř

Prw, Wy WD =

(81)

where we have used p(w,) to denote the unigram probability. The probability con- ditioned on w, is now decomposed by making the independence assumption (Gorin, Levinson, Gertner \ Goldman, 1991) for the w,\,, , $m = 1, ..., M$@ :

a Pr) ae Mon Beye LI ae) (82) n\textasciitildeM\textasciitilde m=]

=consitw,)etw,) [| a Ma=m) 3)

m=

The normalization factor const(w,) is independent of w, and is determined from the normalization constraint: \°Pr(u|w,$\backslash_y$−−ě, −;) = $L. The remarkable property of this model is that the mutual information related term a(w|v)/p$

A word about terminology is in order. The above models are often called adaptive as opposed to static or stationary models. In the case of language modelling, stationarity means that the probability of a word w, is not explicitly dependent on the word position n. In the equation of any of the word association models, there is a position dependence of the probability distribution, but this dependence is only implicit via the positions of the predecessor words w,\,, , $m = 1, ..., 4. Such a type of language model adaptivity must be distinguished from a model like$:

Pe (Wy |e M1) = Po Wal Maem Waaade (84)

where the distribution p,(w,|.) has an explicit dependence on the word position n. Presumably such an explicit dependence is not really needed in language modelling.

Memory weights

The memory weights c„ depend only on the distance m, measured in word units, and can be viewed as coefficients of a window in analogy to windows in signal processing. This window assigns a weight to the interaction between two words according to their position distance m; it can be chosen to be constant or linearly decreasing over position m. A more refined approach would be to use one' half of a bell-shaped window like a Gaussian or a Hamming window. Still another choice is to determine the memory weights from training data w„ a=1,...,N using a maximum likelihood criterion. Assuming the a@(w|v) probabilities to be known and using the method of Lagrange multipliers for the normalization constraint, we have as criterion to be maximized for the model in Equation (76):

Fey stadeES, log [eure \textasciitilde\[Sea |. (85)

„

Setting the partial derivative with respect to c„ to zero, we obtain: Probabilistic dependences in stochastic language modelling 23 SCH | Wn)

1 = —H=0 m=1,...,M. (86) No Ete 1G, |W, 1) a

Multiplying this equation by c„ and using the normalization constraint, we obtain immediately y= 1 and have for the memory weights c„:

os “43 $\backslash atao) en$

man 40H | Wy 1)

The same equation is obtained from the application of the Baum formalism (Jelinek \ Mercer, 1980). Therefore we can use Equation (87) as iterative procedure for calculating the memory weights ¢„ Starting with constant memory weights, the first iteration produces:

(88)

This first iteration computes the memory weight c„ as an average of the normalized interaction factors for the co-occurrence of all word pairs that occur within a distance of m words,

Self-association: cache model

The cache model is introduced by Kuhn and de Mori (1990) can be viewed as a special kind ef word association, where a word is associated with itself. Within certain limits, such a model can adapt itself to changes in word frequencies depending on the topic of the text passage. From this point of view, the widely used M-gram model is virtually stationary because its adaptation is limited by the value of Af, i.e. 4/=1 for bigrams and M=2 for trigrams.

The cache works as a kind of short-term memory by which the probability of the recent M words is increased over the probability of a bigram or trigram model. Formally, the cache model is obtained by choosing the association factors a(.|.) in the form:

(04 | = 0) = (89)

otherwise

We have used the symbol = to denote an equivalence relation. In the simplest case, this equivalence relation requires the words to be identical and we must have: A(w,\,,) = 1.

N. Ney et al.

Another reasonable choice is to consider two words to be equivalent if they differ only in their endings, say the last two characters. The parameter $A(w,\backslash,)mustthenbedeterminedsuchthattheprobabilitiessumuptounityforeachpredecess$

The cache is used in combination with a category-based bigram model. The non-

adaptive membership probabilities are smoothed with the cache component. Each word category is assigned its own memory for the last Mf words.

The cache model is used at the level of unigram probabilities to add an adaptive

component to a non-adaptive language model.

In both variants, a suitable combination of the non-adaptive language model and the cache model is needed, e.g. by using linear interpolation. It should be clear that these two ways of incorporating a cache model can also be used for other types of interaction like the following.

Word interaction: semantic pairs

Next, we consider the word association probabilities $a(w,|w,\backslash,)-Theydescribeasimplemodelofpairwiseinteractions.Foralar$

The log-likelihood as a function of the association probabilities a(w|») in Equation (76) is:

N M

Flea Mod Y tog | E enatvghee md Ea [Dately—1]. 0) n=l ‘m1 , ” The partial derivatives are: OFS spony 2mm OUSWe— me) Baw) OE Negaw]itga ag) A ep

where the symbol d(w;u) is the Kronecker delta which is | for w=u and 0 otherwise. Using the same type of manipulations as for the memory weights, we obtain the following equation:

\(w|v)= const,(w) > S(wyw, a(w| ¥ En nOlViWy- m)

Fant] Hy)!)

where the normalization factor const,(w) does not depend on w, but on v. Probabilistic dependences in stochastic language modelling 25

We do not know whether this equation can be used in an iterative manner as in the case of the memory weights. The Baum framework is not guaranteed to be applicable due to the fact that there are different histories (w,\$_y$...., W, ¯,)$foreachterminthesumoverthewordpositionsn.Ignoringthesepotentialconvergenceproblemsandstartingu$

a(w| v)= const,(w) \5($w; w,$)ěC$mO(V; Wp^{-}mn)$(93)

a=t mat

with another normalization factor constw). The above equation gives roughly the number of co-occurrences of words w and y within a window of size M.

As a direct heuristic approximation for selecting strongly interacting word pairs, we fix a word pair (v,w) and compare its probability with the unigram probability p(w). In this way, we obtain an expression that is related to the mutual information:

I(w,s):= a(9,) log ately). a(v,w) log pron (94)

where a(w,v) is the joint probability of observing the word pair (w,v) within the window m\textasciitildet,..., M. The word association probabilities can be estimated by counting the number of word co-occurrences of w and v within a window of size M. To counteract the sparseness of data, we could assume the symmetry property: a(v,w)=a(w,y).

Another approach is to use the concept of word associations to activate certain subvocabularies depending on the recent M predecessor words. In this approach, the possible values of the association probabilities would be reduced to just ‘yes’ or ‘no’ (or a few more intermediate values) depending on whether there is a significant dependence between the two words under consideration. Such a model for vocabulary activation could have the following form:

— Av, | Wm) 200\where A(w,|w,\,, J£\0, 1\$dependsonlyonthedecisionastowhetherthereisasignificantinteractionbetwee$

Experimental results

In this section, we report on experimental evaluations for the following methods: linear and nonlinear interpolation; statistical clustering for finding word categories; cache model and word association model.

Corpora

Two text corpora are used to train and test the different interpolation schemes and language models: a German (Jugler \ Vehar, 1987) and an English database [the LOB

N. Ney et al.

Tas. I. Statistics of the text data bases (coverage ¢,:= fraction of events seen in training at least \ times)

Corpus German English Type of text newspaper 15 text articles categories corpus size 95691 1157270 vocabulary 14080 49615 number of POS 302 153 Training: Size 71468 865553 n,|size [\unigrams: 9 2 bigrams: si 28 POS bigrams: 4 02 Testing: Size 24203 291697 coverage [\unigrams: ¢, 88 96 bigrams: ¢,/c, 46/36 67/59 POS bigrams: ¢,/e, 96/93 99-8/99-6

corpus, (Kuhn \ de Mori, 1990)]. In both corpora, the vocabulary comprises all full- form words independent of their stems that occur in training and testing. The size of the vocabulary is 14 000 for the German corpus and 50 000 for the English corpus. A non- negligible part of the vocabulary words are not observed in training. The vocabulary comprises also some punctuation marks. The words in both corpora are labelled with

and 153 POSs for the German and the English database, respectively. This labelling

has been done by linguistic experts beforehand. Table I summarizes the characteristic properties of both text corpora and gives also some information on unigram and bigram statistics. The German corpus is fairly homogeneous while the English is of great variety.

Several series of experimental tests were run on the two corpora to measure the performance of various methods for language modelling. In each experiment, the partitioning in training and test data was the same: we used 1/4 of each corpus as test set and the remaining 3/4 as training set.

Interpolation techniques

In the first series of experiments, linear and nonlinear interpolation techniques were compared. We applied the interpolation techniques to the estimation of conditional bigram probabilities. Thus the events under consideration were the words w=1,...,W following a given word y, and the parameters of the nonlinear interpolation model were dependent on y:

p (| y= maxi a D,O + oes a vy) (wv), (96) Probabilistic dependences in stochastic language modelling 27

Taste II. Effect of estimation techniques for nonlinear (a) and linear (b) interpolation on perplexities (word bigrams)

Corpus German English (a) Optimum on test data 650 51 Estimated in training: D, exact 653 543 D exact 653 549 D approx. 633 547 (b) Optimum on test data 707 601 Estimated in training 713 605

where N(v) and N(\v,w) are the counts for word unigrams w and word bigrams (v,w), respectively. The experiments showed that the v-dependence of the parameter D, is not essential. Using just D or b=n,\v)D,/W as independent parameter resulted in the same perplexities. For the optimum of both cases, the nonlinear interpolation results in a 10\improvement over the linear interpolation. Table II gives a comparison of how well the optimal parameters were predicted by the leaving-one-out method applied to the training data. For the case of non-uniform unigram probabilities, D, was estimated by systematically varying it across its range with stepsize 0-05. For constant unigram probabilities, 6 was estimated both by full search and the iterative approximation described earlier. In this case, b was modelled as being independent of y, and thus due to the additivity of the log-likelihood function, all bigram counts were pooled to obtain just one set of \, values. As Table II shows, all these different estimation techniques lead to virtually identical performance on the test data. Even the very crude approximation b=n,/(m, + 2n,) results in the same optimal performance.

The situation is similar for the linear interpolation model. Although here only the approximation formula for the interpolation parameter 4 was used, the corresponding perplexity is optimal, too. Similar results were obtained for the interpolation of the unigram probabilities with a uniform (=“‘zerogram”) distribution, which was necessary since not all words of the test data occur in the training data. Due to its superiority, only the nonlinear interpolation method was used in the subsequent experiments. The dependence of the perplexity on the interpolation parameter is shown in Fig. 1 for the linear and nonlinear interpolation model. Curves Al and A2 show the corpus perplexity on the test data. To study the generalization capability of the methods, we also measured the corpus perplexity on the training data in a leaving-one-out fashion (curves BI and B2). From Fig. 1, we can see that, when going from the training data to the test data, the perplexity goes up by 150, e.g. from 450 to 600. Unlike linear interpolation, nonlinear interpolation produces a rather flat minimum whose position is close to b= t.

Statistical clustering

The clustering algorithm described in Section 4.3 was applied to the training corpus to determine suitable word clusters. The thus obtained word categories were used to define a category bigram model and to compute the perplexity on the test corpus. The number of word categories was varied from 30 to 700. The resulting corpus perplexities are shown in Table III for the German and the English corpus. For comparison purposes,

N. Ney et al.

17080 1100 1000

a

00 700 a

4

aD

| 40

02 04 os os 1

Figure 1. Corpus perplexity on the English corpus as a function of the interpolation parameters: (a) measured on the test data:—Al: linear interpolation; —A2: nonlinear interpolation. (b) measured on the training data with leaving one out: —B1: linear interpolation; —B2: nonlinear interpolation.

TABLE III. Test set perplexity with clustered categories

Number of clusters German English

1=word unigram 1185 1138

674 647

589 561

571 538

557 514

563 500

566 486

576 479

579 478

586 484

$\backslash .541$

= Word bigram 650

$153\backslash_{5}25$

POS 499

the perplexities of the unigram model and of the word bigram models are included. In addition, the perplexities of the POS, i.e. word categories defined by linguistic experts, are shown as well. As can be seen in Table II, the perplexity is drastically decreased by increasing the number of word clusters. Depending on the corpus, we have a flat optimum at about 350 clusters for the German corpus and at 120 for the English corpus. Beyond these numbers, the perplexity increases again slowly toward the word-bigram value. The best clustering result on the English corpus is better than the results for the POS model. On the much smaller German corpus, this is not the case due to the insufficient training data. Probabilistic dependences in stochastic language modelling 29

Taste [V. Most frequent words of some sampie clusters from the English corpus

Cluster A: went turned sat moved walked ran drove stepped bent climbed jumped slipped leaned swung flashed rushed wandered switched rode staggered hurried sank dashed drifted jerked leaped peered bowed slid slowed strode stumbled swam blinked knelt pinned roared sits slumped struggled...

Cluster B: Lord President King Queen Prince Captain Earl General nerve Mayor Major Bishop Princess Temple Count Chief Colonel Commander Admiral High Inspector Madame Mere Albert Arnold Marshal Senator Wood Director Master hydrogen Canon Dante potato Eisenhower Reverend Toulouse-Lautree fullback prosecutor vice...

Cluster C: quickly apart slowly rapidly quietly shortly sharply steadily remote exclusively softly sadly varies eagerly ranging instanily nervously ranged urgently briskly impatiently hurriedly reluctantly calmly smoothly extensively appreciably boldly coldly motionless...

Table IV gives three examples of word categories created by the clustering algorithm. We see that most of the words in each cluster belong to the same syntactic class, namely past tense verbs for cluster A, nouns for cluster B and adverbs for cluster C. Furthermore there are some semantic similarities between the words in a cluster. The majority of the words in cluster A are verbs expressing some kind of motion, and some of the words of cluster B are titles. There are also words which appear to be in the wrong cluster: words like

'potato, Toulouse-Lautrec' are not tities, and words like 'remote, varies' are not adverbs. Although most of the clusters look reasonable, there are also clusters which are difficult to interpret from a linguistic point of view.

Cache model

The influence of the cache model on the perplexity was tested experimentally on the two corpora. For a more detailed analysis of the cache effects, we calculated two partial perplexities PP, and PP, in addition to the usual perplexity PP. The first partial perplexity PP, refers to all words with zero counts (N(w)=0), the second, PP„ covers the words with N(w)>0. This means that the vocabulary contains words that were not seen in the training data, but which are known to occur in the test data. The cache model 'was tested in the two variants mentioned in Section 5.3, namely as part of a membership model in a category bigram model or as part of a word unigram model in a word bigram model. The category-based model made use of the automatically determined word clusters (see Sections 4.3 and 6.3). The size of the cache window was 300 words for the word-based model and 30 words per category for the category-based model. In both cases, the cache model was used in a linear interpolation scheme. Since the results were not very sensitive to the value of the interpolation parameter, the interpolation parameter was adjusted in informal tests by trial and error and then kept fixed during the experiments.

For the category bigram model, Table V summarizes the perplexities without and with a cache for both the German and the English corpus. Comparing the perplexities in Table V, we see that the most obvious result is the difference in improvements obtained for the German and the English corpus. Whereas for the English corpus the perplexity is reduced from 500 to 401 by the cache model, there is only a small improvement from 553 to 520 for the German corpus. This effect is probably caused by the greater heterogeneity of the English corpus: the short-term frequencies of the words differ significantly from

N, Ney et al.

Taste V. Effect of the cache model on perplexities (PP, PP„ PP) using a category bigram model Category bigram German corpus English corpus

Pp, Pp, Pp Pp, Pp, Pp

Without cache 64834 287 553 840646 321 500 With cache 55821 273 520 184155 267 401

TABLE VI. Effect of the cache on perplexities (PP„ PP„ PP) using a word bigram model Word bigram German corpus English corpus

Pp, Pp, Pp Pp, Pp, Pp

Without cache 177860 302 650 3745332 322 S41 With cache 147267 285 607 675254 272 427

their long-term frequencies as estimated on the training corpus. As a result, the adaptation of the language model to the local context is more effective than in the case of the comparatively homogeneous German corpus. The two partial perplexities PP, and PP, show a behaviour similar to the perplexity PP: they are both reduced by the cache model. However, PP„ i.e. the perplexity of words unseen in training, is much more reduced than PP„ in particular for the English corpus.

For the word bigram model, the experiments are summarized in Table VI. The relative improvements brought by the cache model are comparable to the case of the category bigram model although all perplexities are somewhat higher.

Word associations

For this method, we have only preliminary results so far. We used the additive model expressed by Equation (76) and tested it on the English corpus. The word associations a(v,w) were estimated as the relative number

of the co-occurrences of the words v and w within a window of fixed size M, which was shifted through all possible (N-) positions for a training corpus of N words. To rule out short-distance correlations, bigrams and trigrams were removed. To exclude coincidental word co-occurrences, there was a threshold put on the number of words: each of the two words v and w had to occur at jeast three times in order to produce a non-zero word association a(v,w). Typical window sizes ranged from M=10 to =500. The memory weights c„ were found to be of secondary importance only and were constant for ail m.

Table VII shows word pairs that were selected using the following correlation criterion:

$\backslash\backslash_{2}e0)KOopGy"en$

For Table VII, the counts of the word co-occurrences were made symmetrical by ignoring the difference between v...w and w...¥; self-associations were removed. For Probabilistic dependences in stochastic language modelling 31

Taste VIL. Word pairs with high correlation k(v,w) (symmetrical model; no self-associations; window size = 100; minimum number of occurrences N(v)> 100 and N(w)> 100)

vow NW) N(w) N(y,w) k(v,w) food research 178 122 265 54-4 Berlin west 7 149 122 312 commonwealth countries 136 177 163 30:2 cent per 192 380 48) 29-4 desire action 126 129 99 272 German Berlin 157 17 105 25-5 association research 114 122 i) 25-3 changes demand 113 100 63 24-9 Germany — Berlin Wt 1t7 70 24-0 countries trade 177 165 156 23-8 workers union 113 120 69 22:7 hundred feet 105 171 90 22-4 Minister Prime 185 101 92 22-0 agreement United 127 102 6l 210 solution ead 109 124 63 20:8 service health 200 133 122 20-5 building houses 142 129 82 20-0 methods control 125 129 72 19-9 Germany German mw 157 a 197 demand level 100 147 65 19-7 commonwealth United 136 102 58 18-7 values value 105 147 64 18-5 labour party 223 302 279 18-5 marriage married 103 141 60 18-4 countries United 177 102 74 18-3 houses built 129 117 61 18-0 methods method 125 4 57 178 research industry 122 137 66 176 reading books 107 116 49 17-6 section act 133 228 118 17-4 parents area 132 212 107 17-1 Germany west 1 149 63 17-0 President Prime 154 101 59 16-9 children parents 319 132 157 16-6 conditions size 131 105 50 16-2 parents school 132 278 133 16-2 group class 227 101 82 16-0 upon tule 311 110 122 15-9 foreign Berlin 129 7 53 15-7 company union 198 120 83 15-6 commonwealth Africa 136 100 47 15-4 average cent 121 192 9 15-2 commonwealth — Britain 136 206 94 15-0 commonwealth Europe 136 101 46 14-9 German — west 157 149 B 149 wall centre 101 103 34 146 President meeting 184 186 92 143 previous reading 109 107 37 14-2 court rule 134 110 47 142 research council 122 236 88 13-6 surface film 107 120 39 13-6

average per 121 380 140 13-6

Tase VIII. Perplexitics of the word association model (using a minimum number of occurrences: N(v)> 2 and N(w)>2)

(a) Effect of the window length M (for a selection threshold of 0-0005)

Window M Perplexity

1083-2

1048-5

1024-6

1019-1

1024-8

1039-0

10667

unigram 1124-2

(b) Effect of the selection threshold (for a window length A¢= 100)

Number of Threshold word pairs Perplexity

12954071 1004-7

0-00001 11965 694 984-9 0-00002- 11 436 574 981-0

0-00005 10.420 708 O81

0-0001 9354 753 987-6 09-0002 8023 948 1000-1 90-0005 5976911 1019-4 0-001 4 380 703 1032-9 0-002 2852435 1045-4 0-005 1138 296 1061-2

335 069 1072-4

Table VII, the minimum number of word observations was 100. Most of the word pairs shown in Table VII look reasonable from a linguistic point of view. When using the exact local measure of mutual information as selection criterion.

atv,w) P()p(w)”

I(v,w) = a(v,w) log (98)

the highest ranking word pairs seem to be more dominated by text peculiarities like co- occurrences of abbreviations and formulae.

Perplexity measurements using the word association model are shown in Table VIII. For these tests, the selection criterion was the above defined local measure of mutual information. Table VUII(a)\ shows the effect of the window size Mf on the perplexity; the optimal window size is 100. Table VIII(b) shows the effect of the threshold that was put on the mutual information in order to select the relevant word pairs. The word association model was linearly interpolated with unigram model. The interpolation Probabilistic dependences in stochastic language modelling 33

parameters were optimally adjusted on the test data. From the experiments reported above, it is known that this procedure is not critical. In this way, it was possible to study the model of word associations under optimal conditions.

Table VIII shows that the perplexity is reduced from 1124 for the unigram down to

Summary

In this paper, we have studied methods for structuring the probabilistic dependences in stochastic language models. The functional dependences considered are: linear and nonlinear interpolation; M-gram models with equivalence states and word categories; models of pairwise interaction including the cache model.

The nonlinear interpolation method results in significant improvements over linear interpolation in the experimental tests. It has been shown that the leaving-one-out method in combination with the maximum likelihood criterion can be efficiently used for the optimal estimation of interpolation parameters. For the determination of word equivalence classes, an automatic clustering procedure has been designed and successfully tested. Finally, the word association model has been introduced to capture long- distance dependences. The recently introduced cache model can be viewed as a kind of self-association.

Appendices

We present refined approximations for the models of linear and absolute discounting.

Appendix], Linear discounting

It is possible to obtain an approximative solution in the following way. In Equation (32), we subtract a,/N, multiply by (1 — 2) and obtain:

i Nk) a4) (1-2) =4 = 99 (4 Ruaaost MO=1 ” Nalk) For the following, it is convenient to define a parameter A,:

> — MA) (100)

Newest N@)-1 4

Nak)

N, Ney et al.

ws

Mh

ou @-¥i0-9)

aa 4

0H 04 ree) 08 1

Figure A.1. Hlustration of the solution for linear discounting.

Ifa solution with 0<A <1 exists, it is clear from Equation (32) that n,/N is a lower bound for A:

cae

yeish (on The second term in the denominator in the definition of A, is monotonically increasing with 4. Thus using the lower bound A=7,/N, we obtain an upper bound for 4”:

elm (NM: A Any z Wo (102)

A+

Ren(K)> 1

Nak) N-n,

Viewing this quantity 4, as a constant, we have the following quadratic equation for 4

(2-3) (1-2)=Aa,. (103)

This equation can be viewed as the intersection of two function curves that are illustrated in Fig. A.1. By multiplying out and sorting, we obtain:

\-a(1+\

As indicated in Fig. A.1, there are two roots. The root approximating the exact solution is:

a5 (148 -4:) \textasciitilde 1+ 4, nat) (105)

Depending on a suitable choice of a bound for A„ we obtain an upper or lower bound for A, For example, choosing 4„„ as an upper bound for A„ we obtain an upper bound for 2. Fig. A.1 illustrates the solution of Equation (103). The interpolation parameter A is Probabilistic dependences in stochastic language modelling 35

os

oa tb

02

on ana

04 2-VE by 08 1

Figure A.2. Illustration of the solution for absolute discounting.

obtained as intersection point of the parabola 4A(4—n,/N)(1—A) and the function AAA, to which the straight line with the slope 4„/N is an upper bound.

Appendix 2. Absolute discounting Dividing Equation (44) by (1, + 2n,) and defining a new parameter A,:

i

2-b - abe 106 im +2n, z me (106)

Ay: 2 s2 Fol-b

we obtain a quadratic equation for 6, regarding A, as a constant:

,- =a,

(107) (by 6)(2—b) = (1 — BA, (108) BAL + A,)—B(2 + by + A,) + 26)=0. (109)

An argument similar to the case of linear discounting shows that only one of the two roots lies in the open unit interval:

1 base ase thot AD\textasciitilde VO+R + Ay BOI + ADL (110)

It can be shown that by choosing an upper bound for A, this equation results in a lower bound for 6. Anyway, we will show in the following that the above equation is already a very close approximation to the exact solution.

Figure A.2 illustrates how the solution of Equation (107) can be found. The quantity A, has been introduced because it is only weakly dependent on \ and at the same time summarizes the dependence on the measurements n„ r=3,..,.R. For a typical case like

N. Ney et al.

oa 4 392- a9

18)

a 08 0 Os 4 18 a

Figure A.3. Illustration of bounds for absolute discounting.

word bigrams of the LOB corpus (to be exact, the first three quarters), we have (in units of thousand):

In, = 245, 2n, =77, 3n,=45, 4n,=32, 5n,=24, 6n,= 20,

From these measurements, it is easy to verify that for 05<b<1, A, has a variation of less than 8\follows. We use an initial value like b= 0-5 or 0-75 for computing A, an insert this value into Equation (110). The resulting value for 5 will be a tight approximation to the exact solution,

Independent of these experimental observations, we can derive exact lower bounds as follows. A, attains its maximum at b=0:

$=—\backslash!\backslash_z utAySApn + 2n, xr - 1ai)1/68, 10$

nee | ony + t+—n,t... \. 112)

tan (Fs yang) (112)

Tighter bounds might be obtained by constraining the values of b such as 0-5<b<1-0, To handle such constraints, we will use the symbol A„„ rather than A, in the following inequalities.

From Equation (107) and its illustration in Fig. A.2, we see that we need an upper bound also for the function 6+4(1—5)/(2—4). This function attains its maximum at b=2-,/2 and the value of the maximum is (\/2— 1)?<1/5-8.

Thus we have:

bag FA max SPS by. (113)

Another type of lower bound for 5 is obtained by observing the inequality:

W1-B)<(+ Dod), (114)

which can be easily verified by considering the graphic representations of the two

parabolae as shown in Fig. A.3. Both parabolae attain their maximum at 6=0-5. Applying this inequality to Equation (107), we obtain: Probabilistic dependences in stochastic language modelling 37

bb ol4. (Is) and

by—3A,

PPS DK by. (116)

1+ 5A max

It is possible to give a computationally efficient iterative procedure for determining the optimal discounting parameter b. We rewrite Equation (107):

by —b=bes(b) ay) with g(6)-—1— ym toe, (118) nyt2n Sy ‘r\textasciitilde1-b

The function 6g,(b) is monotonically decreasing so that we can use the following iteration for computing d:

penn

+g) (119)

Assuming a suitable starting value, e.g. 6 =1, the convergence of this iteration is guaranteed due to the following inequalities:

b< BOD < pO, (120)

Thus, for the case of absolute discounting this iterative procedure can be viewed to parallel the Baum interative procedure for linear interpolation.

References

Bahl, L.R., Brown, P. F., de Souza, P. V. \ Mercer, R. L. (1989). A tree based statistical language model for natural language speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing,

1001-1008.

Bahl, L. R.. Jelinek, F. \ Mercer, R. L. A maximum likelihood approach to continuous speech recognition, IEBE Transactions on Pattern Analysis and Machine intelligence, 5, 179-190.

Bahl, L.R., Jelinek, F., Mercer, R. L. \ Nadas, A. (1984). Next word statistical predictor. JBM Technical Disclosure Bulletin, 27, TA, 3941-42,

Church, K. W. \ Hanks, °. (1990). Word associations norms, mutual information, and lexicography. Computer Linguistics, 16, 22-29.

Derouault, A.M. \ Merialdo, B. (1986). Natural language modeling for phoneme-to-text transcription. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, 742-749.

Duda, R. 0, \ Hart, P. E. (1973). Pattern Classification and Scene Analysis. Wiley, New York.

Essen, U. \ Steinbiss, V, (1992). Co-occurrence smoothing for stochastic language modelling. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco, CA, pp. I-161-164, March.

N, Ney et al.

Gorin, A.L.„ Levinson, S.E., Gertner, A.N. \ Goldman, E.R. (1991). Adaptive acquisition of language.

Good, IJ. (1953). The population frequencies of species and the estimation of population parameters. Biometrika 49, 237-264.

Jelinek, F. (1985). Markov source modeling of text generation. In The Impact of Processing Techniques on Communication (Skwirzynski, J.K., ed.) Nijhoff, Dordrecht, The Netherlands.

Jelinek, F. (1991). Self-organized language modeling for speech recognition. In Readings in Speech Recognition (Waibel, A., Lee, K-F., eds) Morgan Kaufman Publishers.

Jelinek, F., Mercer, R. \ Roukos, S. (1990). Classifying words for improved statistical language models. IEEE International Conference on Acoustics, Speech and Signal Proceedings, Albuquerque, NM, pp. 621-624, April.

Jelinek, F. \ Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data In Pattern Recognition in Practice (Gelsema, E. S. and Kanal, L.N., eds) pp. 381-397. North-Holland Publ. Company, Amsterdam.

Kaw, S.M. (1980). Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-35, 400-401.

Kugler, M. \ Vehar, M. (1987). Syntaktische klassen fuer das text labelling. Internal report, ESPRIT-Projekt 291/860, Univ. Bochum, Germany.

Kuhn, R. \ de Mori, R. (1990). A cache-based natural language model for speech recognition. [EEE Transactions on Pattern Analysis and Machine Intelligence, 12, 570-583.

Lehmann, E. L. (1983). Theory of Point Estimation, Wiley, New York

Nadas, A. (1984). Estimation of probabilities in the language model of the IBM speech recognition system. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-32, 859-861

Nadas, A. (1985). On Turing's formula for word probabitities. EEE Transactions on Acoustics, Speech and Signal Processing, ASSP-33, 1414-1416.

Rosenfeld, R. \ Huang, X. (1992). Improvements in stochastic language modeling. DARPA Speech and Natural Language Workshop, Harriman, NY, Feb.

(Received 9 October 199? and accepted 12 July 1993)