Variable-length category-based *n*-grams for language modelling

T.R. Niesler and P.C. Woodland

CUED/F-INFENG/TR.215

30 April 1995

Cambridge University Engineering Department

Trumpington Street, Cambridge, CB2 1PZ

trn@eng.cam.ac.uk
pcw@eng.cam.ac.uk

Abstract

This report concerns the theoretical development and subsequent evaluation of *n*-gram language models based on word categories. In particular, part-of-speech word classifications have been employed as a means of incorporating significant amounts of *a-priori* grammatical information into the model. The utilisation of categories diminishes the problem of data sparseness which plagues conventional word-based *n*-gram approaches, and therefore yields a fundamentally more compact model. Furthermore, it allows the use of larger *n*, and a strategy by means of which successively longer *n*-grams are selectively added to the model according to a cross-validation likelihood criterion is proposed. This enables the model compactness to be maintained while allowing longer range effects to be modelled where they benefit performance. The language modelling approach was applied to the LOB corpus in order to assess its effectiveness. When compared with models of corresponding complexity constructed according to conventional *n*-gram methods, it is found that the proposed procedures render language models exhibiting superior performance. Furthermore, comparison with word-based *n*-gram models shows that comparable performance may be achieved at a large reduction in model size. An ultimate aim of the described work is to construct language models from very large text corpora, the contents of which are generally not annotated with the required part-of-speech classifications. For this reason the use of the category-based language model as a statistical tagger is introduced as a means of automatically determining this information, and is shown by means of tests on the LOB corpus to yield very good tagging accuracies.

Contents

1. Introduction

This report describes the development of an *n*-gram language model based on word categories. Word categories are groupings of individual words, where a word will be considered to be defined completely by its spelling. While a word-based *n*-gram model bases its statistics on the observed frequencies of the words themselves, its category-based counterpart makes use of the observed frequencies of the word categories. In doing so, the latter approach has the following advantages with respect to the former :

- Since the number of different categories is much smaller than the number of different words, the category *n*-gram counts in the training set will be less sparse than word *n*-gram counts.
- Since there are fewer different category *n*-grams, the model will be more compact and thus occupy less memory.
- Since the training set is less sparse and the model is more compact, the use of values of *n* larger than 3 or 4 becomes feasible both from a statistical as well as a storage viewpoint. The use of deeper contexts has been seen to have a marked impact on the quality of the language model [Shannon 50], [Bahl 89].
- Being based on word categories, the model is able to generalise to unseen word sequences. Furthermore, new words may be added to the lexicon of the language model without having to gather further *n*-gram statistics. Only the categories to which the new entries belong need be known, since their sequential behaviour may be expected to be captured by the existing language model *n*-gram statistics.

Furthermore, in choosing the categories to correspond to the grammatically meaningful **part-of-speech (POS)** classifications, the language model is implicitly set the goal of modelling the syntactic patterns of the text. An *n*-gram model topology has been chosen, and although this cannot capture the complete syntactic structure of the language [Chomsky 56], it has proven to be a very successful approximation in practice, a phenomenon that may be attributed to the often significantly local syntactic constructs in English text [Jelinek 90]. Furthermore, the construction of *n*-gram models is quite computationally inexpensive, a quality which becomes important when treating large quantities of text, as is the case in this work.

The LOB corpus [Johansson 86] was chosen as a starting point, as it is tagged using a fairly detailed set of POS word classifications. Since POS-tagged text corpora are in general small in size (LOB contains around 1.1 million words), and an aim of this work is to derive language models from the much larger untagged corpora currently becoming available, some method of determining the required word POS classifications is called for. The approach taken here is to employ the language model derived from the LOB corpus as the basis for a statistical tagger, with which the untagged text is consequently processed, thereby effectively using the information extracted from the smaller LOB corpus to bootstrap the language model building process on the larger body of text.

The following sections describe the development of such an *n*-gram POS-based language model, and present experimental results to provide an indication of its performance on the LOB corpus.

2. Notational conventions

The following conventions for referring to sequences of events will be used consistently throughout the remaining text. A sequence of *N* consecutive events is denoted by

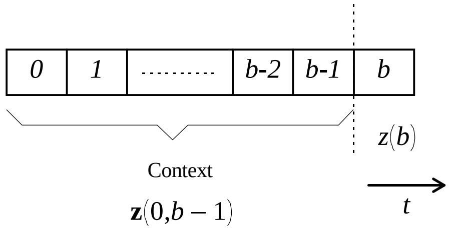
$$\mathbf{z}(0, N-1) \equiv \{z(0), z(1), \ \dots \ , z(N-1)\}$$

A particular segment of this sequence may be referred to by indicating the appropriate starting and ending indeces. For example,

$$\mathbf{z}(a,b) \equiv \{z(a), z(a+1), \ \dots \ , z(b-1), z(b)\}.$$

where $0 \leq a \leq b \leq N-1$. The sequences are assumed to be ordered temporally, meaning that the rightmost element of a sequence is also the most recent.

The **context** of an event will be taken to refer to its immediate history. The context of $z(b)$, for example, is $\mathbf{z}(0,b-1)$, as illustrated in the following figure.



The particular identity of a member of such a sequence is indicated by means of a subscripted index. For example, assuming that the $z(i)$ are drawn from an alphabet of size *K*, we have :

$$z(i) \in \{z_0, z_1, \ \dots \ , z_{K-1}\}$$

and when $z(i) = z_j$ this is denoted by

$$z_j(i)$$

The number of times a sequence $\mathbf{z}(a,b)$ occurs within a certain corpus will be denoted by

$$N(\mathbf{z}(a,b)).$$

Finally, the following symbols will consistently be used to refer to particular types of sequences :

w : Sequence of words.

v : Sequence of word categories.

Therefore $\mathbf{w}(a,b)$ refers to the subsequence of $(b - a + 1)$ words drawn from the sequence **w** and starting at position *a* in **w**.

3. Theoretical development

3.1 Structure of the language model

Let the relationship between a word and its category be defined by the mapping :

$$v_j = G(w_i) \qquad j \in 0,1,\ \dots\ ,N_{wc}-1 \qquad \text{...... (1)}$$

where v_j is the category to which w_i is assigned by the operation *G*, and N_{wc} corresponds to the number of different word categories. Note that *G* is in general one-to-many since a word may belong to more than one category simultaneously. When the categories are syntactic POS groupings, for example, this occurs because a word may have more than one grammatical function.

> ***First structural assumption*** : The probability of occurrence of a word will be assumed to be dependent solely upon the category to which it is believed to belong, and will thus be written as $P(w_i|v_j)$.

Now let each word history $\mathbf{w}(0,b)$ be classified into particular equivalence class s_i by means of the mapping operator *S* :

$$s_i = S(\mathbf{w}(0,b)) \qquad i \in 0,1,\ \dots\ ,N_{hc}-1 \qquad \text{...... (2)}$$

where N_{hc} is the number of history equivalence classes. This definition is very general, and for the particular language model under study a history equivalence class will be defined to be a POS category sequence associated with a portion of the most recent word history, i.e. an *n*-gram of POS categories :

$$\begin{aligned} S(\mathbf{w}(0,b)) &= \mathbf{v}(a,b) \\ &= \{v(a), v(a+1), \dots, v(b)\}\ ,\ \text{where}\ \ v(i) \in G(w(i))\ \ \text{and}\ \ 0 \le a \le b \end{aligned} \qquad \text{...... (3)}$$

Similar approaches in which the history mapping is also based on word categories have been described in [Brown 92], [Ney 94] and [Kuhn 90]. Note that, since *G* is in general one-to-many, *S* will also be one-to-many, and therefore a particular word sequence $\mathbf{w}(0,b)$ may map to more than one history equivalence class.

> ***Second structural assumption*** : The probability of witnessing a particular category *v*(*i*) will be assumed to be dependent only upon its category *n*-gram context, and will thus be written as $P(v(i)|\mathbf{v}(i-k,i-1))$, where $k > 0$.

Using equation (1) and the first structural assumption, the language model probability may be calculated as follows :

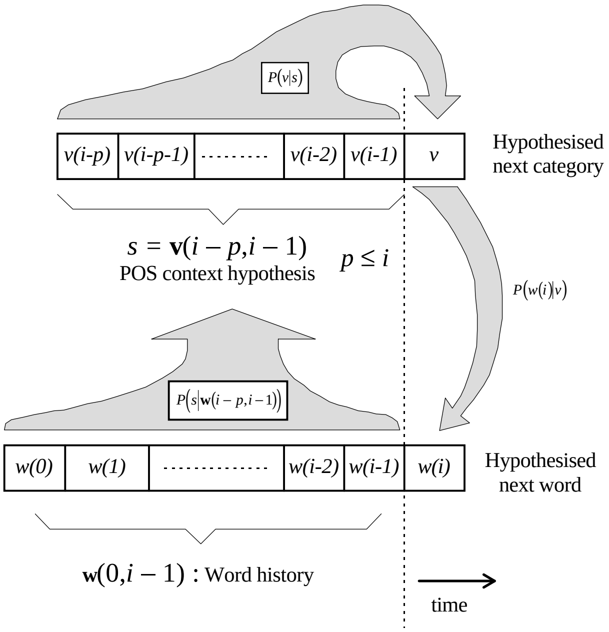
$$P(w(i)|\mathbf{w}(0,i-1)) = \sum_{\forall\ v:v\in G(w(i))} P(w(i)|v) \cdot P(v|\mathbf{w}(0,i-1)) \qquad \text{...... (4)}$$

This equation computes the probability of the next word *w*(*i*) given the word history $\mathbf{w}(0,i-1)$. To do so it makes use of $P(w(i)|v)$, the probability that *w*(*i*) is the next word given that *v* is the next POS category, and $P(v|\mathbf{w}(0,i-1))$, the probability that *v* is the next POS category given that the word history is $\mathbf{w}(0,i-1)$. The summation takes all categories for which the former probability is nonzero into account, recalling that the word *w*(*i*) may belong to more than one category *v*.

Using equation (2) and the second structural assumption, the second term on the right hand side of equation (4) may be further decomposed as follows :

$$P\left(v\middle|\mathbf{w}(0,i-1)\right)=\sum_{\forall\ s:s\in S\left(\mathbf{w}(0,i-1)\right)} P\left(v\middle|s\right)\cdot P\left(s\middle|\mathbf{w}(0,i-1)\right) \qquad \text{...... (5)}$$

where $P\left(v\middle|s\right)$ is the probability that *v* is the next POS category given that the word history **w**(0,i – 1) belongs to equivalence class *s*, and $P\left(s\middle|\mathbf{w}(0,i-1)\right)$ is the probability that the equivalence class of **w**(0,*i* – 1) is indeed *s*. As before, the summation accounts for all the possible history equivalence classes of the word history **w**(0,*i* – 1) for which the latter probabilities will be nonzero. The interrelation of the three component probabilities of equations (4) and (5) is illustrated in the following figure, and the subsequent sections treat the estimation of each individually.

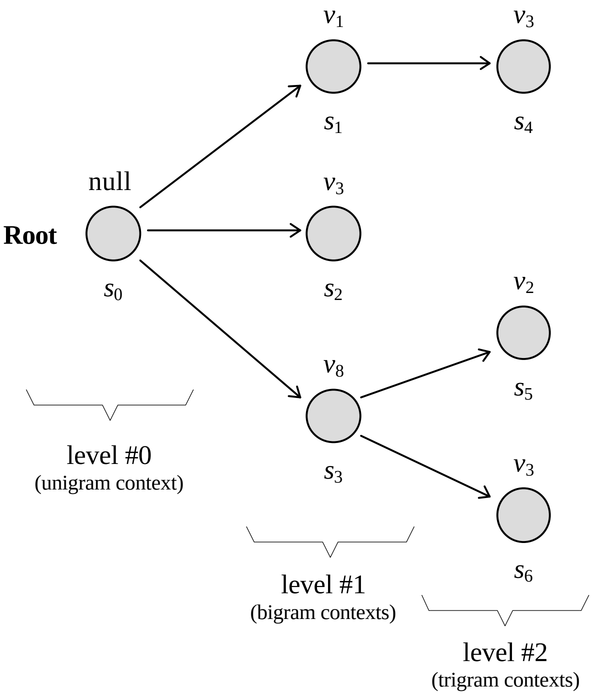


3.2 Estimating $P(v_j|s_m)$

In order to store the POS *n*-grams compactly, a tree data-structure is employed in which each node is associated with a particular word category and in which paths originating at the root correspond to category *n*-grams. From definition (3) this implies that each node represents a distinct history equivalence class[1] s_m, and therefore has associated with it a conditional probability density function $P(v|s_m)$. By not restricting the length of the individual paths through the tree, contexts of arbitrary depth are catered for. The following figure illustrates this structure by means of an example. Nodes are labelled both with the specific history equivalence class s_m they represent, as well as the category defining the *n*-gram with respect to the parent node. In particular, for this example, the history equivalence class s_1 corresponds to the bigram

[1] The set of all nodes therefore constitutes the set of all possible equivalence classes.

context $\mathbf{v}(i-1,i-1)=\{v_1\}$ and the history equivalence class s_5 to the trigram context $\mathbf{v}(i-2,i-1)=\{v_2,v_8\}$.



The structure described above is merely an efficient way of storing *n*-gram statistics. Assuming that the counts have been determined, the probabilities $P(v|s_m)$ are estimated by application of Katz's back-off [Katz 87] in conjunction with Ney's nonlinear discounting scheme [Ney 94]. However, before counting may be carried out, it is necessary to specify both the structure of the tree as well as the POS category associated with each node. Let the notion of a *level* within a tree refer to the depth of the context under consideration, as shown in the figure. The approach taken in conventional *n*-gram modelling is simply to count the occurrences of all events in all contexts $\mathbf{v}(i-p,i-1)$ found in the training set such that $p < n$. This will be termed the *standard* *n*-gram model-building technique.

Instead of this exhaustive counting scheme, a means of selecting only those contexts useful from a language model point of view was desired so as to ensure model compactness by avoiding the exponential growth in the tree size when taking increasingly longer contexts into account. To achieve this the following level-by-level growing strategy is employed.

1. **Initialisation** : $L = -1$
2. $L = L+1$
3. **Grow** : Add level #L to level #$(L-1)$ by adding all the $(L+1)$-grams occurring in the training set for which the L-grams already exist in the tree.
4. **Prune** : For every (newly created) leaf in level #L, apply a quality criterion, and discard the leaf if it fails.
5. **Termination** : If there are a nonzero number of leaves remaining in level #L, goto step 2.

The chosen pruning criterion determines whether the addition of the node contributed to the likelihood of the training set to a significant extent. When calculating the training set likelihood, however, it is necessary to employ some method of cross-validation to prevent the likelihood figure from increasing monotonically as the context lengths increase. The leaving-one-out framework [Duda 73] was employed for this purpose. In particular, referring to appendix A, the leaving-one-out log likelihood function may be written as

$$LL_{\text{cum}}\left(\Omega^{\text{tot}}\right) = \sum_{i=0}^{N-1} \log\left(P\left(v(i) \middle| ctxt\left(v(i)\right), \Omega_i^{\text{RT}}\right)\right)$$

where N is the number of words in the entire training corpus Ω^{tot}, $ctxt\left(v(i)\right)$ is the context within which $v(i)$ is found in Ω^{tot}, and $P\left(v(i) \middle| ctxt\left(v(i)\right), \Omega^{\text{RT}}\right)$ is the probability estimated by the n-gram model obtained from Ω^{RT}, the retained part formed by removal of the heldout part Ω^{HO} from Ω^{tot}. Assuming that the contexts corresponding to each node are labelled as $s_0, s_1, \ldots, s_{N_n-1}$, N_n being the number of nodes in the tree, this likelihood may be rewritten as the sum of the contributions of each node :

$$\begin{aligned} LL_{\text{cum}}\left(\Omega^{\text{tot}}\right) &= \sum_{n=0}^{N_n-1}\left(\sum_{v(j):ctxt\left(v(j)\right)=s_n} \log\left(P\left(v_j \middle| s_n, \Omega_j^{\text{RT}}\right)\right)\right) \\ &= \sum_{n=0}^{N_n-1} LL_{\text{cum}}^{s_n} \end{aligned}$$

where

$$LL_{\text{cum}}^{s_n} = \sum_{k=0}^{N_{vv}} N_{s_n}\left(v_k\right) \cdot \log\left(P\left(v_k \middle| s_n, \Omega_k^{\text{RT}}\right)\right) \qquad \text{...... (6)}$$

and $N_{s_n}\left(v_k\right)$ is the number of times v_k was seen in context s_n in the training set Ω^{tot}, N_{vv} is the number of different categories, and Ω_k^{RT} is the retained part formed when $\Omega_k^{\text{HO}} = v_k$. Equation (6) allows the contribution node s_n makes to the total leaving-one-out likelihood to be computed. Now assume that node s_n is a leaf, and that the change in likelihood resulting from the addition of a child $s_{n+\varepsilon}$ should be calculated. While s_n refers to the original parent node, s_n' is used to denote this node after the addition of the child. The change in likelihood is then given by

$$\Delta LL_{\text{cum}}^{s_n} = LL_{\text{cum}}^{s_n'} + LL_{\text{cum}}^{s_{n+\varepsilon}} - LL_{\text{cum}}^{s_n} \qquad \text{...... (7)}$$

The quantity $LL_{\text{cum}}^{s_n}$ may be precalculated for the context s_n, while $LL_{\text{cum}}^{s_n'}$ and $LL_{\text{cum}}^{s_{n+\varepsilon}}$ must be evaluated for each candidate child $s_{n+\varepsilon}$ using equation (6). In terms of these quantities, the pruning criterion is

$$\Delta LL_{\text{cum}}^{s_n} > \Delta_L \;\; ? \qquad \text{...... (8)}$$

where

$$\Delta_L = -\lambda \cdot LL_{\text{cum}}\left(\Omega^{\text{tot}}\right) \qquad \text{...... (9)}$$

This requires the addition of the new node to lead to a likelihood increase of at least a threshold Δ_L, where this value is defined to be a fraction λ of the total likelihood so as to make the choice of the threshold fairly problem independent.

3.2.1 POS perplexity

The probability estimate $P\left(v \middle| s\right)$ may be used to calculate a perplexity figure given a sequence of categories. Since the categories are taken to be POS word classifications, the perplexity calculated in this way will be referred to as the POS-perplexity. It gives an indication of the average branching factor of the sequence of word categories, and will be used to quantify the performance in isolation of the category n-gram language model component in much the same way as the word-perplexity is used to gauge the quality of the language model as a whole.

3.3 Estimating $P(w_i|v_j)$

In estimating $P\left(w_i\middle|v_j\right)$, it was assumed that each category v_j had been witnessed sufficiently many times in the training set to allow the application of the relative frequency estimate :

$$P\left(w_i\middle|v_j\right)=\frac{N\left(w_i\middle|v_j\right)}{N\left(v_j\right)} \qquad \text{...... (10)}$$

Due to the finite number of words in the language model lexicon, it is inevitable that out-of-vocabulary (OOV) words will be encountered while processing new text. As it is useful for the language model to be able to estimate the probability of occurrence of such unknown events[2], a procedure by means of which it may be estimated using the leaving-one-out cross-validation strategy has been adopted, and is described in detail in appendix B. It results in the addition of a dedicated "UW" entry to the lexicon, the counts of which are estimated for each category individually, thereby allowing the calculation of the OOV probability by straightforward application of (10).

3.4 Estimating $P(s_m|\mathbf{w}(0,i-1))$

The purpose of this component of the language model is to estimate the probability that a particular word history $\mathbf{w}(0,i-1)$ corresponds to the POS context s_m. Since a word may have multiple POS classifications, there are in general many possible contexts to which $\mathbf{w}(0,i-1)$ could belong. The set of such contexts as well as the probabilities associated with each may be calculated by means of a recursive approach, in which we assume the contexts and their probabilities to be known for $\mathbf{w}(0,i-1)$, and derive the corresponding results for $\mathbf{w}(0,i)$. First define :

$\mathbf{v}_j^{\text{hyp}}(a,b)$: A possible classification of the word sequence $\mathbf{w}(a,b)$ into POS categories (termed a ***hypothesis*** hereafter, individual hypotheses being distinguished by the index *j*)

$s_m=F_{\text{tree}}(\mathbf{v}(a,b))$: The history equivalence class corresponding to the deepest match of the POS category sequence $\mathbf{v}(a,b)$ within the *n*-gram tree. Although each sequence $\mathbf{v}(a,b)$ has a unique history equivalence class s_m , many different $\mathbf{v}(a,b)$ may map to the same s_m .

$Depth\left(F_{\text{tree}}(\mathbf{v}(a,b))\right)$: The depth of the above match. Unigrams correspond to a depth of zero (null context), bigrams to a depth of one, trigrams two, and so on.

$N_{H(a,b)}$: The number of hypotheses for the word sequence $\mathbf{w}(a,b)$.

For each possible hypothesis we require the probability that it is the correct classification of the word string. Denote this probability by $P\left(\mathbf{v}_j^{\text{hyp}}(a,b)\middle|\mathbf{w}(a,b)\right)$, so that :

$$\sum_{j=0}^{N_{H(a,b)}-1} P\left(\mathbf{v}_j^{\text{hyp}}(a,b)\middle|\mathbf{w}(a,b)\right)=1 \qquad \text{...... (11)}$$

[2] In particular, this has an important effect when employing the language model as a statistical tagger.

The approach in this development will be to determine expressions for $P\left(\mathbf{v}_j^{\text{hyp}}(0,i)\middle|\mathbf{w}(0,i)\right)$, from which the desired probability of the history equivalence class may be obtained easily :

$$P\left(s_m\middle|\mathbf{w}(0,i)\right)=\sum_{\forall j:F_{\text{tree}}\left(\mathbf{v}_j^{\text{hyp}}(0,i)\right)=s_m} P\left(\mathbf{v}_j^{\text{hyp}}(0,i)\middle|\mathbf{w}(0,i)\right) \qquad \text{...... (12)}$$

The explicit maintenance of the hypotheses is necessary (as opposed to simply keeping a list of the most likely history equivalence classes) due to the varying lengths of the *n*-grams. In particular, it may occur that

$$Depth\left(F_{\text{tree}}\left(\mathbf{v}(0,i-1)\right)\right)<Depth\left(F_{\text{tree}}\left(\mathbf{v}(0,i)\right)\right)-1$$

in which case the *n*-gram probability estimate based on **v**(0,*i*) makes use of more contextual information than is implicit in the history equivalence class $F_{\text{tree}}\left(\mathbf{v}(0,i-1)\right)$. [3]

For the calculation of the probabilities in (12), it is in practice only necessary to maintain a set of hypotheses **v**(*i* – *D*,*i* – 1) of depth *D* such that *D* equals or exceeds the maximum length of any *n*-gram stored in the tree, i.e.:

$$D\geq\max_{\forall\mathbf{v}}\left(Depth\left(F_{\text{tree}}(\mathbf{v})\right)\right) \qquad \text{...... (13)}$$

This guarantees that the hypothesis is always at least as deep as any path through the tree. Identical hypotheses arising during this process (having differed only in elements *v*(*i* – α) for α > *D*, may be merged by summing their probabilities.

Given a set of existing hypotheses $\left\{\mathbf{v}_j^{\text{hyp}}(0,i-1)\right\}$, the set of new hypotheses is $\left\{\mathbf{v}_j^{\text{hyp}}(0,i-1),v_k\right\}$ for all (*j*,*k*) such that $j=\left\{0,1,\ \ldots\ ,N_{H(0,i-1)}-1\right\}$ and $k=\left\{0,1,\ \ldots\ ,N_{vv}-1\right\}$ where N_{vv} is the number of different POS categories. Consider now the particular postulate $\mathbf{v}_{j'}^{\text{hyp}}(0,i)=\left\{\mathbf{v}_j^{\text{hyp}}(0,i-1),v_k\right\}$, the prime over the index indicating that there is in general no fixed relation between the ordering of the two sets of hypotheses.

Using Bayes rule, we may write :

$$\begin{aligned}P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i)\middle|\mathbf{w}(0,i)\right)&=\frac{P\left(\mathbf{w}(0,i)\middle|\mathbf{v}_{j'}^{\text{hyp}}(0,i)\right)\cdot P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i)\right)}{P\left(\mathbf{w}(0,i)\right)}\\&=\frac{P\left(\mathbf{w}(0,i),\mathbf{v}_{j'}^{\text{hyp}}(0,i)\right)}{P\left(\mathbf{w}(0,i)\right)}\end{aligned} \qquad \text{...... (14)}$$

but, recalling the first structural assumption from section 3.1 it follows that :

$$\begin{aligned}P\left(\mathbf{w}(0,i)\middle|\mathbf{v}_{j'}^{\text{hyp}}(0,i)\right)&=\prod_{k=0}^{i}P\left(w(k)\middle|v_{j'}^{\text{hyp}}(k)\right)\\&=P\left(w(i)\middle|v_{j'}^{\text{hyp}}(i)\right)\cdot P\left(\mathbf{w}(0,i-1)\middle|\mathbf{v}_{j'}^{\text{hyp}}(0,i-1)\right)\end{aligned} \qquad \text{...... (15)}$$

and, from the second structural assumption (the *n*-gram model), we find that

[3] It is for this reason that the language model cannot be treated as a first-order Markov process with the history equivalence classes as states, since the implication of this statement is that the transition probability may depend on more than just the identities of the emitting and receiving states.

$$\begin{aligned} P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i)\right) &= \prod_{k=0}^{i} P\left(v_{j'}^{\text{hyp}}(k) \middle| F_{tree}\left(\mathbf{v}_{j'}^{\text{hyp}}(0,k-1)\right)\right) \\ &= P\left(v_{j'}^{\text{hyp}}(i) \middle| F_{tree}\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i-1)\right)\right) \cdot P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i-1)\right) \end{aligned} \qquad \text{...... (16)}$$

where $\mathbf{v}_{j'}^{\text{hyp}}(0,-1)$ is the single initial null hypothesis and $F_{tree}\left(\mathbf{v}_{j'}^{\text{hyp}}(0,-1)\right)$ the associated unigram context, so that $P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,-1)\right) = 1$. From (14), (15) and (16) it follows that

$$P\left(\mathbf{w}(0,i), \mathbf{v}_{j'}^{\text{hyp}}(0,i)\right) = P\left(w(i) \middle| v_{j'}^{\text{hyp}}(i)\right) \cdot P\left(v_{j'}^{\text{hyp}}(i) \middle| F_{\text{tree}}\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i-1)\right)\right) \cdot P\left(\mathbf{w}(0,i-1), \mathbf{v}_{j'}^{\text{hyp}}(0,i-1)\right) \qquad \text{...... (17)}$$

Finally, note that

$$P\left(\mathbf{w}(0,i)\right) = \sum_{j'=0}^{N_{H(0,i)}} P\left(\mathbf{w}(0,i), \mathbf{v}_{j'}^{\text{hyp}}(0,i)\right) \qquad \text{...... (18)}$$

At any given instant, the most likely postulate is that for which $P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i) \middle| \mathbf{w}(0,i) \right)$ is a maximum. Due to the large number of *n*-grams held in the tree and the constraint (13), the number of hypotheses becomes extremely large as *i* increases, and it is necessary to restrict storage to the $N_H^{\max}$ most likely candidates by choosing those for which this probability is greatest. This implies that valid hypotheses may be discarded, in which case (11) will no longer be satisfied, i.e.

$$\sum_{q=0}^{N_H^{\max}} P\left(\mathbf{v}_q^{\text{hyp}}(0,i) \middle| \mathbf{w}(0,i)\right) < 1$$

where $\mathbf{v}_q^{\text{hyp}}(0,i)$ is taken in this case to refer to the q_{th} most likely hypothesis. The language model (5), however, requires these probabilities to sum to unity. By replacing equation (18) with

$$P\left(\mathbf{w}(0,i)\right) = \sum_{q=0}^{N_H^{\max}} P\left(\mathbf{w}(0,i), \mathbf{v}_q^{\text{hyp}}(0,i)\right) \qquad \text{...... (19)}$$

these conditional probabilities are renormalised on application of equation (14). In effect the probability mass associated with the discarded hypotheses is distributed proportionally among those which are retained. Note that, since according to equation (14) the quantity $P\left(\mathbf{w}(0,i)\right)$ is common to all new hypotheses, the choice of the $N_H^{\max}$ best candidates may be made by considering the joint probabilities $P\left(\mathbf{w}(0,i), \mathbf{v}_{j'}^{\text{hyp}}(0,i)\right)$ instead of the conditional probabilities $P\left(\mathbf{v}_{j'}^{\text{hyp}}(0,i) \middle| \mathbf{w}(0,i)\right)$.

The complete recursive procedure is summarised below. It is assumed that the set of $N_H^{\text{old}} \leq N_H^{\max}$ best previous hypotheses $\mathbf{v}_j^{\text{hyp}}(0,i-1)$ as well as the corresponding probabilities $P\left(\mathbf{w}(0,i), \mathbf{v}_j^{\text{hyp}}(0,i-1)\right)$ are available in arrays collectively referred to as $\mathbf{H}^{\text{old}}$. Similarly, the set of $N_H^{\text{new}} \leq N_H^{\max}$ updated context hypotheses with their corresponding probabilities $P\left(\mathbf{w}(0,i), \mathbf{v}_j^{\text{hyp}}(0,i)\right)$ and $P\left(\mathbf{v}^{\text{hyp}}(0,i) \middle| \mathbf{w}(0,i)\right)$ will be stored in $\mathbf{H}^{\text{new}}$. Initialisation is accomplished by setting *i* = –1 and placing a single null hypothesis in $\mathbf{H}^{\text{new}}$, i.e. $v_0^{\text{hyp}}(0) = \text{null}$ and $N_H^{\text{new}} = 1$.

```
1.  Copy all $\mathbf{v}^{\text{hyp}}(0,i)$ and corresponding $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ in $\mathbf{H}^{\text{new}}$ to $\mathbf{H}^{\text{old}}$
2.  Clear $\mathbf{H}^{\text{new}}$
3.  $i = i + 1$
4.  For each hypothesis $\mathbf{v}_j^{\text{hyp}}$  $j = \{0,1,\ \dots\ ,N_H^{\text{old}} - 1\}$ in $\mathbf{H}^{\text{old}}$
5.      For each POS category $v_k$ such that $w(i) \in v_k$
6.          $\mathbf{v}^{\text{hyp}}(0,i) = \{\mathbf{v}_j^{\text{hyp}}, v_k\}$
7.          Calculate $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ using (17).
8.          If $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ is greater than any of the entries in $\mathbf{H}^{\text{new}}$, insert $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ and $\mathbf{v}^{\text{hyp}}(0,i)$ into $\mathbf{H}^{\text{new}}$, possibly overwriting the smallest entry in the process.
9.  Calculate $P\big(\mathbf{w}(0,i)\big)$ using (19).
10. Calculate $P\big(\mathbf{v}^{\text{hyp}}(0,i) \big| \mathbf{w}(0,i)\big)$ for each $q = \{0,1,\ \dots\ ,N_H^{\text{new}} - 1\}$ in $\mathbf{H}^{\text{new}}$ using (14).
11. $\mathbf{H}^{\text{new}}$ now contains the set of best new hypotheses a well as the corresponding probabilities $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ and $P\big(\mathbf{v}^{\text{hyp}}(0,i) \big| \mathbf{w}(0,i)\big)$. Use (12) to calculate $P\big(s_m \big| \mathbf{w}(0,i)\big)$.
```

3.4.1 Incorporating a beam search into hypothesis maintenance

The above procedure maintains a fixed maximum number of hypotheses for the word history $\mathbf{w}(0,i)$. Often a significant number of these have very low associated $P\big(\mathbf{v}_q^{\text{hyp}}(0,i) \big| \mathbf{w}(0,i)\big)$ values. By discarding such unlikely hypotheses the computational efficiency of the procedure may be improved considerably. In particular only that set of hypotheses with associated probabilities that are at least a certain fraction[4] of the probability to corresponding the most likely hypothesis are maintained.

Letting $P_{\mathbf{w},\mathbf{v}}^{\max}(i)$ denote the maximum $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ entry in $\mathbf{H}^{\text{new}}$, the condition under which a certain hypothesis is maintained may be stated as :

$$P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big) \geq \delta \cdot P_{\mathbf{w},\mathbf{v}}^{\max}(i) \qquad \text{......} \ (20)$$

In practice this means that step 8 in the above procedure must be reformulated as follows :

```
8a. If (20) is satisfied by the new hypothesis, insert $P\big(\mathbf{w}(0,i),\mathbf{v}^{\text{hyp}}(0,i)\big)$ and $\mathbf{v}^{\text{hyp}}(0,i)$ into $\mathbf{H}^{\text{new}}$, possibly overwriting the smallest entry in the process.
8b. Remove from $\mathbf{H}^{\text{new}}$ any hypothesis for which (20) fails.
```

The last step ensures that any hypothesis in $\mathbf{H}^{\text{new}}$ will be discarded that no longer satisfies (20) due to the subsequent addition of more likely hypotheses.

[4] i.e.: falling within the beam.

The chief motivation for introducing this beam-search scheme is that it may be used to trade accuracy for computational efficiency of the algorithm, an issue that becomes particularly important when the language model is used to tag large quantities of text.

3.4.2 Employing the language model as a POS tagger

Since the language modelling approach described in this work assumes the availability of POS information for each word in the training corpus, and since such information is only available in certain corpora of limited size, an automatic means of annotating untagged text with the POS classifications is required for large, newly available corpora to be used in language model construction.

When using the language model to tag text, the aim is to find the most likely[5] POS assignment for each word in a given sentence. Denoting the sentence by $\mathbf{w}(0,N-1)$, this corresponds to finding that sequence $\mathbf{v}(0,N-1)$ for which the probability

$$P\big(\mathbf{v}(0,N-1)\big|\mathbf{w}(0,N-1)\big)$$

is a maximum. Recalling that a list of these probabilities as well as their corresponding POS hypotheses are maintained by the procedure described earlier in section 3.4, it is evident that the calculation of the language model probability component $P\big(s\big|\mathbf{w}(0,i)\big)$ implicitly involves a tagging operation. In particular, it maintains a list of the most likely sequences of POS assignments for the sequence of words $\mathbf{w}(0,i)$ with respect to the language model statistics.

Since it is assumed that the POS *n*-gram model does not operate across sentence boundaries, sentences may be tagged one at a time. Therefore, with reference to the procedure described in section 3.4, the tagging process entails the following steps for each sentence :

1. Initialise $\mathbf{H}^{\text{new}}$, set $i = -1$.
2. Execute steps 1 – 11 for each word of the current sentence in turn.
3. The hypothesis $\mathbf{v}(0,N-1)$ with the highest $P\big(\mathbf{v}(0,N-1)\big|\mathbf{w}(0,N-1)\big)$ is the most likely sequence of tags for the sentence.

It is in the context of text tagging that the beam-search hypothesis maintenance procedure detailed in the previous section will be employed, since the large size of the untagged corpora demands a computationally efficient algorithm so as to allow the tagging to be carried out in a reasonable amount of time.

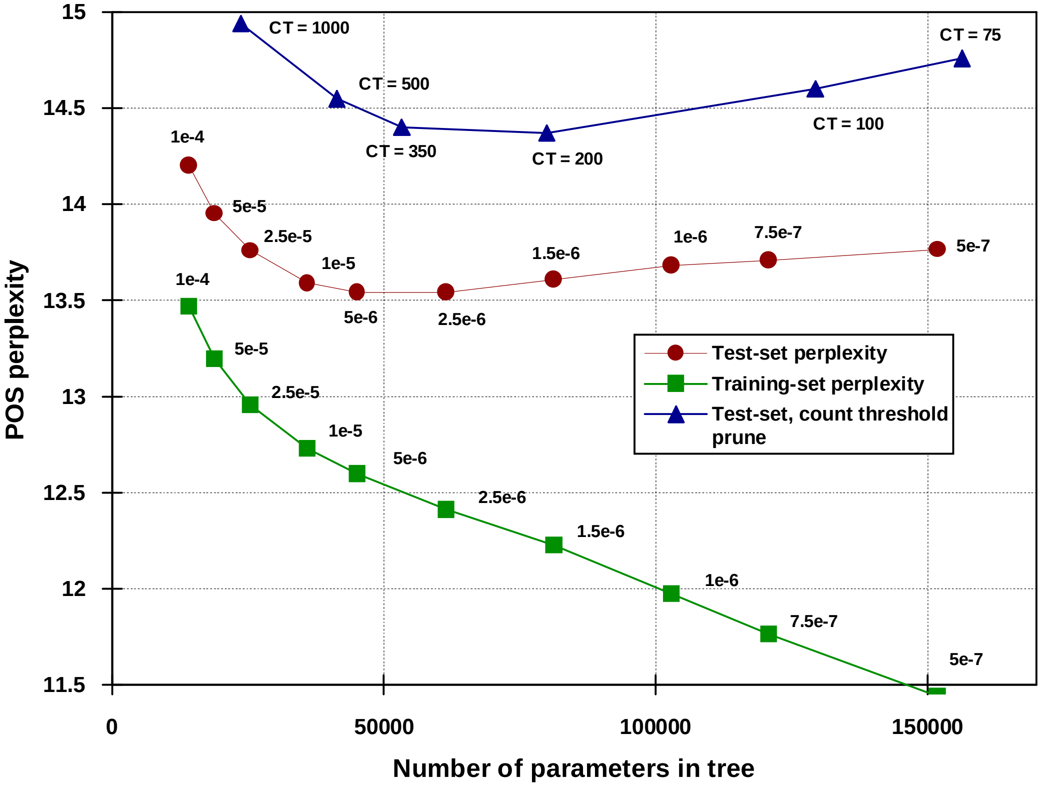
[5] with respect to the language model's structure, and its *n*-gram and class membership statistics.

4. Experimental results

In this section the performance of the language model construction and application techniques described in the preceding section will be assessed by experimental evaluation on the LOB corpus [Johansson 86], which consists of approximately 1.1 million words of POS-tagged English text chosen from a variety of sources, including newspaper reportage, fiction and scientific writing. The results include both language model perplexities and complexities, as well as tagging accuracies.

4.1 Constructing POS *n*-gram trees

Using the method of section 3.2, POS *n*-gram language model trees were constructed from the LOB corpus using various pruning thresholds. The tree complexities[6] as well as language model POS perplexities[7] for various threshold values are shown by the lower two curves in the following graph, where each point has been labelled with the corresponding threshold value. In addition to this, the test-set perplexities obtained when pruning is achieved simply by thresholding the total number of occurrences of an event in the training-set are shown for various choices of this threshold (termed a *count threshold* and abbreviated by "CT"). Note that this technique is a commonly employed to make *n*-gram models more compact.



From the above figure it is evident that :

- The training set perplexity decreases as the number of parameters in the tree increases. This is to be expected, since the pruning criterion disallows a reduction in the training set likelihood.
- As the complexity of the tree increases, the test-set perplexity moves through a global minimum. The initial decrease may be ascribed to underfitting of the data due to an insufficient number of parameters, and the subsequent increase to

[6] The total number of parameters in a tree has been taken as a measure of its complexity
[7] Refer to section 3.2.1.

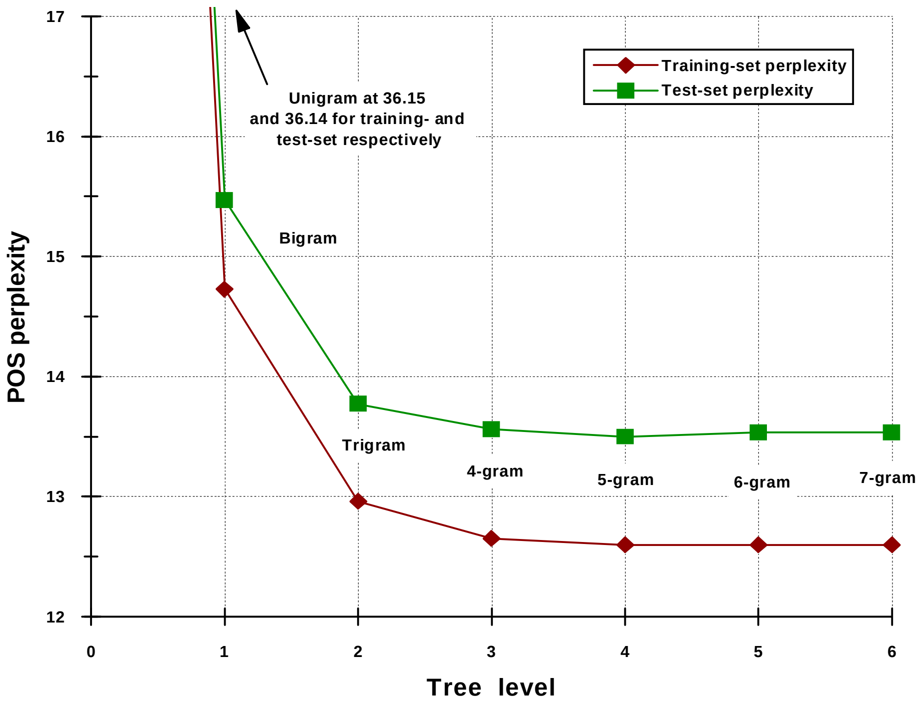
overfitting of the data due to an excessive number of parameters in the language model tree. Overfitting occurs despite the use of leaving-one-out cross-validation, although its effect has been significantly reduced in comparison with the use of the count-threshold pruning technique.

- The optimal tree (λ = 5e-6) is substantially more compact than the tree grown with λ = 0, which has 550725 parameters and a test-set perplexity of 13.91 (not shown on graph), and has a significantly lower perplexity than a tree of comparable size obtained by pruning according to count-thresholds.

Considering the tree obtained with λ = 5e-6, the number of nodes found in each level is presented in the following table. Recall that the number of nodes in level L corresponds to the number of $(L+1)$-grams in the language model. Note that as n increases, the data becomes more sparse, and consequently the probability estimates based thereupon become more unreliable, the cross-validation likelihood criterion leads to a reduction in the number of *n*-grams added to the language model.

Level	Number of nodes
0 (Unigram)	1
1 (Bigram)	123
2 (Trigram)	1001
3 (4-gram)	543
4 (5-gram)	124
5 (6-gram)	17
6 (7-gram)	1
TOTAL	1810

Furthermore, the training- and test-set perplexities were seen to evolve as follows with the addition of each tree level. The test-set perplexity is seen to increase slightly after the addition of the 5th and 6th levels. This is due to the approximate way in which the leaving-one-out framework models cross-validation with the test-set.



4.2 Language model word-perplexities

Three trees were constructed using $\lambda = 5\text{e-}6$, and were subsequently used in the language model described in section 3.1. The first two are bigram and trigram structures, obtained by stopping tree growth beyond levels 1 and 2 respectively. The third was obtained by allowing the tree-growing algorithm to execute to completion, and is termed a **varigram** structure due to the varying lengths of the *n*-grams it contains. Each POS category was augmented with the unknown word UW, the counts of which were calculated according to the estimates presented in appendix B. OOV words were excluded from the perplexity calculations, but included by means of the generic UW in *n*-gram contexts.

The following table shows the word perplexities obtained for each tree when varying the number of maintained history postulates $N_H^{\max}$. The beam selection of section 3.4.1 was not employed in perplexity calculations.

	Number of hypotheses			
	1	2	4	10
Bigram	671.3	610.2	604.1	603.2
Trigram	634.7	555.2	545.2	544.1
Varigram	629.3	548.9	536.7	534.1

The word perplexities are thus seen to decrease monotonically as the number of hypotheses is increased, demonstrating that the history equivalence class ambiguity has a significant effect on the language model performance. The largest decrease occurs as $N_H^{\max}$ is increased from 1 to 2, further increments leading to smaller reductions. The figures in the table indicate that values of $N_H^{\max}$ in the range 5 ... 10 will yield near-optimal results. Furthermore, the longer contexts in the varigram tree lead to a drop in perplexity with respect to the bigram and trigram structures.

A word-based trigram language model for the same corpus achieves a perplexity of 474 but contains 986892 parameters. Therefore a 11.3 % decrease in perplexity is accompanied by an almost 22-fold increase in the number of parameters. The improved performance of the word-based model may be attributed to its ability to model statistical dependencies between particular words. While the category-based model does not have access to this information, it may compensate by offering improved generalisation where the training set is sparse.

4.3 Tagging text with the language model

An ongoing aim of this work is to construct language models from bodies of text that are much larger than the LOB corpus but are in general not annotated with POS information. For this reason the performance a statistical tagger based on the LOB–trained POS language model is of interest, since it would provide a means of obtaining the required POS classifications.

4.3.1 Tagging accuracy of the varigram tagger

The varigram language model trained on the LOB training-set (95%) was used to tag the test-set (5%) by means of the procedure described in in section 3.4.2, and the result compared with the tags in the test set to determine the tagging accuracy. The corresponding figures (i.e. employing the same training- and test-set) for the ACQUILEX tagger [Elworthy 93] are provided as a benchmark.

	ACQUILEX	**varigram**
Overall tagging accuracy	94.03	95.13
Tagging accuracy of known words	95.77	96.31
Tagging accuracy of OOV words (2.51%)	31.17	49.30

These results show that the performance of both taggers is quite similar, but that the varigram tagger exhibits a considerable improvement in the error rate when tagging OOV words. This difference is attributed to both the longer *n*-gram contexts used, as well as the method by means of which the probabilities of unknown words are calculated (refer to appendix B).

4.3.2 Lexicon augmentation

The results of the preceding section show that the tagging accuracy for OOV words is significantly lower than for words which do in fact appear in the tagger's lexicon. For this reason the effect of augmenting the lexicon with words from various additional information sources was investigated. In particular, the following sources were employed :

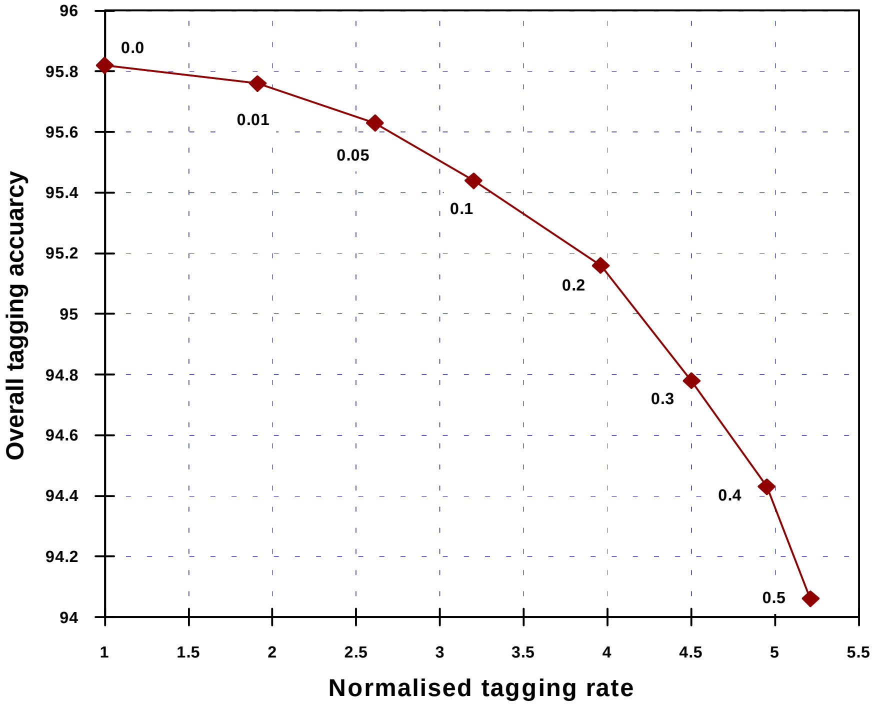
- Word spellings and POS assignments from the Oxford Advanced Learner's Dictionary (available electronically). The mapping used to convert the dictionary's POS classifications to those employed by the lexicon is described in appendix C.
- A list of 5000 frequent names and surnames. These were included since the OOV words were seen to include a high proportion (approx. 70%) of proper nouns.
- Genitive cases of words already present in the lexicon in their standard but not their genitive form.

The following table shows the tagging accuracies as well as OOV rates for language models built using the augmented as well as the unaugmented lexica. It is apparent that the OOV rate has more than halved, and that the overall tagging accuracy has improved. Note that, due to the larger vocabulary, the language model using the augmented lexicon actually has a higher perplexity than that using the unaugmented lexicon when tested on the test-set, but due to the large difference in tagging accuracy between known and OOV words, the consequent drop in the OOV rate nevertheless leads to an improvement in the overall tagging accuracy.

	No augmentation	With augmentation
OOV rate	2.51 %	1.05 %
Overall tagging accuracy	95.13	95.82
Tagging accuracy of known words	96.31	96.26
Tagging accuracy of OOV words	49.30	54.55

4.3.3 Introducing the beam search

The beam-search mechanism described in section 3.4.1 allows computational complexity to be traded for tagging accuracy by limiting the number of hypotheses maintained during the tagging operation. The following graph illustrates how these two quantities are affected by the choice of the beam parameter δ. A normalised tagging rate of unity corresponds to a real tagging rate of approximately 433 words per second on an HP 735 workstation.



A choice of δ = 0.1 leads to a factor 3.2 speed improvement at the expense of a 9.1% increase in tagging error rate.

5. Conclusion

A category-based language model employing *n*-grams of varying lengths has been described. A procedure by which the *n*-grams may be chosen so as to optimise the model compactness with respect to its performance has been presented, and experiments using the LOB corpus show language models constructed in this way to outperform conventional *n*-gram approaches. A word-based trigram model for the same corpus offers a 11.25% perplexity reduction at the expense of an almost 22-fold increase in model complexity, thus making the category-based model a strong contender where compactness is of prime importance. Finally, the use of the category-based language model as a statistical tagger has been described, and when tested on the LOB corpus it exhibits somewhat improved accuracy when compared with a standard fixed-length *n*-gram tagger.

6. References

[Bahl 89] : Bahl, L. , Brown, P.F. , de Souza, P.V. , Mercer, R.L. ; *A tree-based statistical language model for natural language speech recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing , vol. 37, no. 7, July 1989.

[Brown 92] : Brown, P.F. , de Souza, P.V. , Mercer, R.L. , Della Pietra, V.J. , Lai, J.C. ; *Class-based n-gram models of natural language*, Computational Linguistics, Vol. 8, no. 4, 1992.

[Chomsky 56] : Chomsky, N. ; *Three models for the description of language*, IRE transactions on information theory, vol. IT-2, Proceedings of the Symposium on Information Theory, 1956.

[Duda 73] : Duda, R.O. , Hart, P.E. ; *Pattern classification and scene analysis*; Wiley, New York, 1973.

[Elworthy 93] : Elworthy, D. ; *Tagger suite user's manual*, May 1993.

[Jelinek 90] : Jelinek, F. ; *Up from trigrams*. In : Readings in speech recognition. Waibel, A. , Lee, K.F. (eds.). Morgan Kaufman, San Mateo, California, 1990.

[Johansson 86] : Johansson, S. , Atwell, R. Garside R. , Leech G. ; *The Tagged LOB corpus user's manual* ; Norwegian Computing Centre for the Humanities, Bergen, Norway 1986.

[Katz 87] : Katz, S.M. ; *Estimation of probabilities from sparse data for the language model component of a speech recogniser*; IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 35, no. 3, March 1987, pp. 400 - 401.

[Kuhn 90] : Kuhn, R. , de Mori, R. ; *A cache-based natural language model for speech recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 6, June 1990.

[Ney 94] : Ney, H. , Essen, U. , Kneser, R. ; *On structuring probabilistic dependencies in stochastic language modelling*, Computer Speech and Language, vol. 8, pp. 1-38, 1994.

[Shannon 50] : Shannon, C.E. ; *Communication theory : exposition of fundamentals*, IRE Transactions on Information Theory, no. 1, February 1950.

7. Appendix A : Leaving-one-out cross-validation

Consider a training set Ω^{tot} containing N members that is divided into two subsets, Ω^{RT} (termed the retained part) and Ω^{HO} (termed the heldout part). Conventional cross-validation approaches estimate some parameters from Ω^{RT} by optimising the model performance over Ω^{HO}. When the parameters are part of a probability estimate, this is done by maximising the likelihood of Ω^{HO}.

The leaving-one-out approach [Duda 73] is an extension of this procedure in which Ω^{HO} is chosen to contain exactly one member of Ω^{tot}, while Ω^{RT} consists of the remaining $N - 1$. Let Ω^{tot} consist of the events $\{x(0), x(1), \ldots, x(N-1)\}$ where each $x(i)$ is drawn from a finite alphabet $\mathbf{A}_x = (x_0, x_1, \ldots, x_{N_A-1})$, where N_A is the alphabet size. Denoting the single member in Ω^{HO} by $x(i)$, the log likelihood of the heldout part is indicated as :

$$\begin{aligned} LL\left(\Omega^{\text{HO}}, \Omega^{\text{RT}}\right) &= LL\left(x(i), \Omega^{\text{RT}}\right) \\ &= \log\left(P\left(x(i), \Omega^{\text{RT}}\right)\right) \end{aligned}$$

where the probability estimate $P(\cdot)$ is a function of Ω^{RT} since it is made exclusively on the grounds of the data in the retained part. Leaving-one-out cross-validation involves the consideration of all N possible ways in which Ω^{tot} may be partitioned into Ω^{HO} and Ω^{RT}. Denote the N partitions formed by assigning $x(i)$ to Ω^{HO} by Ω_i^{HO} and Ω_i^{RT} , where $i = \{0,1, \ldots, N-1\}$. The cumulative log likelihood over each of these partitions is then

$$\begin{aligned} LL_{\text{cum}}\left(\Omega^{\text{tot}}\right) &= \sum_{i=0}^{N-1} LL\left(x(i), \Omega_i^{\text{RT}}\right) \\ &= \sum_{i=0}^{N-1} \log\left(P\left(x(i), \Omega_i^{\text{RT}}\right)\right) \end{aligned} \qquad \text{...... (1)}$$

Now denote the number of occurrences of x_i in Ω^{tot} by $N^{\text{tot}}(x_i)$, this may be rewritten as

$$LL_{\text{cum}}\left(\Omega^{\text{tot}}\right) = \sum_{i=0}^{N_v-1} N^{tot}\left(x_i\right) \cdot \log\left(P\left(x(i), \Omega_i^{\text{RT}}\right)\right) \qquad \text{...... (2)}$$

By making additional assumptions regarding the form of $P\left(x(i), \Omega_i^{\text{RT}}\right)$, this expression may be simplified further.

In making use of all possible subdivisions into retained and heldout parts, the leaving-one-out approach makes optimal use of the available training data, an important consideration in situations where the data is sparse, as is indeed the case for language modelling problems. Its drawback is the increased computation implied by the exhaustive partitioning operation, although efficient results may sometimes be obtained by simplifications possible for specific choices of $P\left(x(i), \Omega_i^{\text{RT}}\right)$.

References

[Duda 73] : Duda, R.O. , Hart, P.E. ; *Pattern classification and scene analysis*; Wiley, New York, 1973.

8. Appendix B: Dealing with unknown words

In open vocabulary tasks, the language model will in general always encounter words not present in its vocabulary. The occurrence of such words, generally referred to as **out-of-vocabulary** (or simply **OOV)**, may be modelled by adding the dedicated entry “UW” to the lexicon. As this entry represents not a particular but any OOV word, it is in fact a word class with an unknown number of members. By means of the UW entry, it will be possible for the language model both to estimate the probability of occurrence of an unknown word, as well as to use UW occurrences as part of its context.

Since the language model is derived from the training corpus which also defines the vocabulary, the UW event itself is not witnessed during training, and it is therefore necessary to estimate the probability of seeing an OOV wordexplicitly for every category v_j .[8] In order to do this, the leaving-one-out cross-validation framework has been employed.

Let :

v_j be the word category for which we would like to estimate $P(\mathrm{UW} \mid v_j)$.

N_c be the total number of words in the training set (i.e. the corpus size).

Now consider the N_c possible ways in which the training corpus may be split into the two partitions $\mathbf{W}_{\mathrm{i}}^{\mathrm{RT}}$ and $w_{\mathrm{i}}^{\mathrm{HO}}$, where the first contains $N_c - 1$ members and the second exactly one, and where $i = \{0,1, \ldots ,N_c-1\}$. Denote the category to which the word w_i belongs in the training set by $cat_{trn}(w_{\mathrm{i}})$ and define :

$$\delta\left(w_i^{\mathrm{HO}}, v_j\right) = \begin{cases} 1 \text{ if } \left(w_i^{\mathrm{HO}}{}_i \notin \mathbf{W}_i^{\mathrm{RT}}\right) \cap \left(cat_{trn}\left(w_i^{\mathrm{HO}}\right) = v_j\right) \\ 0 \text{ otherwise} \end{cases}$$

Then the probability $P(\mathrm{UW}|v_j)$ of encountering an unknown event in category v_j within the sub-corpus $\mathbf{W}_{\mathrm{i}}^{\mathrm{RT}}$ of size $N_c - 1$ may be estimated by the relative frequency :

$$\boxed{P\left(\mathrm{UW}\middle|v_j\right) = E\left\{\delta\left(w_i^{\mathrm{HO}}, v_j\right)\right\} = \frac{\sum_{i=0}^{N_c-1} \delta\left(w_i^{\mathrm{HO}}, v_j\right)}{N\left(v_j\right)}} \qquad \text{...... (B1)}$$

where $N(v_j)$ is the total number of events that has been seen in category v_j . Note that the numerator is simply the sum of the number of events occurring in this category that also occur only once in the entire corpus, and may thus be determined by means of a simple counting operation.

The number of unknown events $N_{\mathrm{uw}}(v_j)$ that may be expected to be seen in category v_j in a sub-corpus of size $N_c - 1$ may be estimated from the relative frequencies in a similar fashion :

$$P\left(\mathrm{UW}\middle|v_j\right) = \frac{N_{\mathrm{uw}}\left(v_j\right)}{N\left(v_j\right) + N_{\mathrm{uw}}\left(v_j\right)}$$

$$\Rightarrow \boxed{N_{\mathrm{uw}}\left(v_j\right) = \frac{P\left(\mathrm{UW}\middle|v_j\right) \cdot N\left(v_j\right)}{1 - P\left(\mathrm{UW}\middle|v_j\right)}} \qquad \text{...... (B2)}$$

This equation may be used to estimate the count that should be assigned to the unknown event in every category v_j . Precautions must be taken, however, when the training data for certain contexts is sparse. In these circumstances it may

[8] More precisely, we estimate the probability of witnessing OOV words in a body of text that exhibits the same statistical behaviour with respect to the language model extracted from the training corpus, but that has not formed part of this training set in any way.

happen that the numerator of (B1) approaches or even equals the denominator, leading to an extremely large estimate for $N_{\text{uw}}(v_j)$ according to (B2). In order to avoid this, the probability estimate (B1) has been altered heuristically as follows :

$$\boxed{P\left(\text{UW}\middle|v_j\right) = E\left\{\delta\left(w_i^{\text{HO}}, v_j\right)\right\} = \frac{\sum_{i=0}^{N_c-1} \delta\left(w_i^{\text{HO}}, v_j\right)}{N\left(v_j\right)+\eta} \qquad \text{where} \quad \eta > 0} \qquad \text{...... (B3)}$$

The constant η ensures that the denominator is always larger than the numerator, thus never permitting $P(\text{UW}|v_j) = 1$. When $N(v_j)$ is small, indicating the category v_j to be sparsely trained, η will have a significant effect on $P(\text{UW}|v_j)$. However, as $N(v_j)$ increases, η becomes less significant and the estimate (B3) approaches (B1). Intuitively the quantity η may be interpreted as an indication of the number of observations that must have been made in a category v_j for relative frequency estimates made within it to be used with confidence. The effect η on the language model performance was seen empirically to be weak, and a values in the range 5 - 10 were found to yield satisfactory results for the LOB corpus.

It is important to realise that the addition of the unknown event to the vocabulary does not interfere with the estimation of *n*-gram probabilities using Good-Turing or discounting approaches, since the former addresses the occurrence of un*known* events in a certain context, while the latter is concerned with the estimation of probability for un*seen* events, i.e. events that have been seen individually but not in conjuction with each other in an *n*-gram sense.

It is interesting to note that, when v_j is assumed to contain the entire training corpus, the estimate (B1) becomes :

$$P\left(\text{UW}\middle|v_j\right) = P(\text{UW}) = \frac{C_1}{N_c}$$

which is the Good-Turing estimate for the probability of unseen events [Good 53], [Katz 87].

References

[Good 53] : Good, I.J. ; *The population frequencies of species and the estimation of population parameters,* Biometrika vol. 40, pp. 237 - 264, 1953.

[Katz 87] : Katz, S.M. ; *Estimation of probabilities from sparse data for the language model component of a speech recogniser*; IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 35, no. 3, March 1987, pp. 400 - 401.

[Nádas 84] : Nádas, A. ; *Estimation of probabilities in the language model of the IBM speech recognition system*, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 32, no. 4, August 1984, pp. 859 - 861.

9. Appendix C: OALD tag mapping

An electronic version of the Oxford Advanced Learner's Dictionary (OALD) containing sufficiently detailed word tagging information was used to augment the lexicon extracted from the LOB corpus with the purpose of reducing the OOV rate. The following table lists the OALD tags used in this process, as well as the LOB tags to which they were mapped. In a few cases it was necessary to map an OALD tag to more than one LOB tag, this being indicated by separating the latter with colons.

OALD tag	LOB tag
Gb	VBG
Gc	VBD
Gd	VBN
Ha	VBZ
Hb	VBG
Hc	VBD
Hd	VBN
H0	VB
H1	VB
H2	VB
H3	VB
H4	VB
H5	VB
Ia	VBZ
Ib	VBG
Ic	VBD
Id	VBN
I0	VB
I1	VB
I2	VB
I3	VB
I4	VB
I5	VB
Ja	VBZ
Jb	VBG
Jc	VBD
Jd	VBN
J0	VB
J1	VB
J2	VB
J3	VB
J4	VB
J5	VB

OALD tag	LOB tag
Ki	NN
Kj	NNS
K6	NN
K7	NN
K8	NN
K9	NN : NNS
Lk	NN
L@	NN
Mi	NN
Mj	NNS
M6	NN
M7	NN
M8	NN
M9	NN : NNS
M@	NN
Nl	NP
Nm	NP
Nn	NP
No	NP
OA	JJ
OB	JJ
OC	JJ
OD	JJ
OE	JJ
Op	JJ
Oq	JJB
Or	JJR
Os	JJT
Ot	JJ
Pu	RB
P+	RP
T-	IN
W-	UH