ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Speech and Language 17 (2003) 87–104

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Language modelling for Russian and English using words and classes

E.W.D. Whittaker*, P.C. Woodland

Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, UK

Received 24 October 2001; received in revised form 6 September 2002; accepted 9 September 2002

Abstract

This paper examines statistical language modelling of Russian and English in the context of automatic speech recognition. The characteristics of both a Russian and an English text corpus of similar composition are discussed with reference to the properties of both languages. In particular, it is shown that to achieve the same vocabulary coverage as a 65,000 word vocabulary for English, a 430,000 word vocabulary is required for Russian. The implications of this observation motivate the remainder of the paper. Perplexity experiments are reported for word-based N-gram modelling of the two languages and the differences are examined. It is found that, in contrast to English, there is little gain in using 4-grams over trigrams for modelling Russian. Class-based N-gram modelling is then considered and perplexity experiments are reported for two different types of class models, a two-sided model and a novel, one-sided model for which classes are generated automatically. Word and class model combinations show the two-sided model results in lower perplexities than combinations with the one-sided model. However, the very large Russian vocabulary favours the use of the one-sided model since the clustering algorithm, used to obtain word classes automatically, is significantly faster. Lattice rescoring experiments are then reported on an English-language broadcast news task which show that both combinations of the word model with either type of class model produce identical reductions in word error rate.

1. Introduction

This paper considers language modelling for automatic speech recognition of Russian and English. The problems that are encountered with the modelling of Russian are found to be

* Corresponding author. Present address: Philips Research Laboratory, Weisshausstrasse 2, 52066, Aachen, Germany. Tel.: +49-173-2525-145. *E-mail address:* edward@ewdw.com (E.W.D. Whittaker).

0885-2308/03/$ - see front matter
doi:10.1016/S0885-2308(02)00047-5

significantly different to those observed when modelling English. For example, it is shown that to achieve the same vocabulary coverage as a 65,000 word vocabulary for English, a 430,000 word vocabulary is required for Russian. A comparison is made wherever possible between the statistical characteristics of the two languages, based both on data from a Russian and an English text corpus and also in terms of the performance of language modelling techniques that are applied to both languages. One consequence of the different characteristics of Russian was that novel modelling techniques were needed. In this paper, we investigate the performance, on both the Russian and English data, of two different class-based language models that use automatically derived classes. One, referred to as the two-sided class model, has already received much attention in the literature. The other, referred to as a one-sided model, is not known to have been investigated before and its use was motivated by the particular properties of Russian that were encountered. The new model trades a small loss in the ability to generalise with a significantly improved clustering speed over the conventional class model. Further details of this work are given in Whittaker (2000) together with experiments using sub-word particle language modelling techniques.

In Section 2, we outline the major differences between Russian and English that are likely to affect statistical language modelling of Russian. This is followed in Section 3 with the description of the Russian and English corpora used in the experimental work in this paper and by an overview and comparison of their salient characteristics. In Section 4, experimental results are reported for conventional word-based N-gram models on the two corpora and in Section 5 experimental results are reported for two different types of class-based language model. Section 6 reports word error rate results obtained from lattice rescoring experiments for the word models and the two different types of class models on an English language broadcast news task.

2. Russian vs. English

There are two important differences between Russian and English that are of relevance to statistical language modelling and that are shared to varying degrees by many other languages: word formation and word ordering. Russian words typically exhibit clearer morphological patterns than can be found in English words. For example, a Russian word will often contain the following, easily identifiable, constituent parts: a *root* which can be thought of as responsible for the nuclear meaning of the verb, attached to which may be zero or more *derivational prefix*(*es*) and zero or one *suffix*, which together form a *stem*. The *stem* often acquires an entirely new lexical meaning with the presence of these affixes. Of most relevance to language modelling, however, is the *inflection* (*inflectional suffix*), which is appended to the stem and which determines the grammatical case (of which there are six), gender (masculine, feminine, or neuter), number, etc. of the *word*. The presence of the inflection results in many more different word forms representing the same word than is the case for the equivalent English word, for which there are generally no more than two distinct forms, typically with and without an appended "s." The direct consequence of this rich morphology is that the coverage of a Russian vocabulary will tend to be significantly less than that of the same sized English vocabulary.

English compensates for having less grammatical information encoded within the words themselves, by imposing strict constraints on the relative order of words in a sentence. In the

sentence, "*The boy kicks the ball.*", it is only clear who is doing what to whom from the order in which the words are written. In Russian, on the other hand, the subject and object of the sentence can only be determined by each word's inflection and by agreement with the verb, not from the order of the words themselves. In fact, the above sentence translated into Russian, could be expressed with the six (there is no definite article in Russian) different permutations of the Russian for the three words "boy", "kicks" and "ball" without loss of meaning. Clearly, this phenomenon has the potential for seriously weakening the predictive power of N-gram language models, however, in reality some word orderings are preferred stylistically to others. In particular, a different emphasis is placed on a word depending on its position in the sentence, so the permutations of a sequence of words will actually occur with different frequencies.

3. The corpora and their characteristics

Two language modelling corpora, one for Russian and one for English, were required for the language modelling experiments presented in this paper. Both corpora needed to be sufficiently similar in terms of composition and size, so that their characteristics and the experimental results could be compared sensibly. Two similar corpora were eventually located, each of which contained around 100 million tokens after textual preprocessing.

3.1. The Russian corpus

At the time of writing, there are still no large, commercially available Russian language text corpora. However, a large source of Russian text material was eventually located in Russia and this source was used as the basis for all the Russian language modelling experiments contained in this paper. This corpus of Russian texts is very varied in content, ranging from classical literature and translations of popular foreign novels to lists of anecdotes and jokes. After the corpus had been cleaned and the character mappings normalised, sentence boundary information (sentence-start and sentence-end markers) was added which replaced various punctuation markers such as full-stops, ellipsis, and exclamation marks. All other punctuation was removed. Finally, an important procedure in corpus preparation was also executed – the removal of repeated "chunks" of text from the corpus, including whole articles or excerpts from other texts where necessary. If such repetitions were to occur both in the training set and the test set then spurious results (most likely to be optimistic results) would be obtained. An heuristic method of determining identical repetitions in the corpus of fifty-word sequences was developed and repeated segments up to and including the nearest sentence-end marker were removed from the corpus. The final stage of corpus preprocessing mapped all numerical digits to a ⟨NUMBER⟩ symbol since there is no simple method for converting numerical digits into word representations in Russian, because each number changes its grammatical case, gender, and number depending on its role in the sentence.

3.2. The British National Corpus

Since it was desired to conduct similar experiments on English to those on Russian, the choice of English language corpus, of which there were many, was dictated by the characteristics of the

Russian corpus, over which there was little control. It was decided that the most similar, readily available English language corpus in terms of its composition and size, was the British National Corpus (BNC corpus) (Burnard, 1995). The BNC corpus is a collection of English language texts ranging from *belles lettres* and entire novels to transcriptions of spoken language. As with the Russian corpus, there were large sections of repeated data in the BNC corpus. The same method, mentioned above, of removing repeated fifty-word sequences was also used to produce a "cleaner" English corpus. After text normalisation had been performed and to maintain consistency with the processing of the Russian corpus, numerical digits were all mapped to a ⟨NUMBER⟩ symbol. The occurrence of these numerical digits, however, was significantly lower than in the Russian corpus.

3.3. Corpus partitioning

Once the corpora had been cleaned and normalised, they were both partitioned into one training and two test sets: one development set (`dev-test`) with which to optimise the parameters of the model (which are estimated using the training set) and one evaluation set (`eval-test`) with which to evaluate the performance of the language model. Partitions were made in the approximate ratio of 98:1:1 for `training`: `dev-test`: `eval-test` set sizes using multiples of five contiguous sentences to give partitions that were as homogeneous as possible. The size of the resulting test sets was found to be sufficient for providing accurate perplexity results.

3.4. Vocabulary growth and corpus size

It is evident from the elementary introduction to differences between Russian and English given in Section 2 that the number of different words encountered in the two languages will differ significantly. What is unknown is to what extent this difference manifests itself and the consequences it will have on existing language modelling techniques. In Fig. 1, the growth in the number of unique tokens (essentially the vocabulary size) is plotted against corpus size for the two corpora.

It is observed that the rate of growth of the vocabulary for Russian is approximately two and a half times greater than that for English. This is perhaps surprisingly low, since Russian words generally have many more inflected forms than words in English. One possible explanation is that not all inflected forms of a Russian word are used with the same frequency. The other interesting observation from this graph is that the vocabulary size is nowhere near saturating with the increasing corpus size. Unfortunately, there was no accurate means of determining Russian word stems so as to examine their growth with corpus size.

3.5. Coverage and vocabulary size

A useful measure of the coverage of a vocabulary is the percentage of words that are encountered in some held-out text which are *out-of-vocabulary* (OOV). The OOV-rate is thus defined here as the number of tokens in some held-out text that are not in the vocabulary, divided by the total number of tokens in the text. The significance of a particular vocabulary's OOV-rate on the recognition performance of a speech recogniser cannot be understated. If a word is not in the

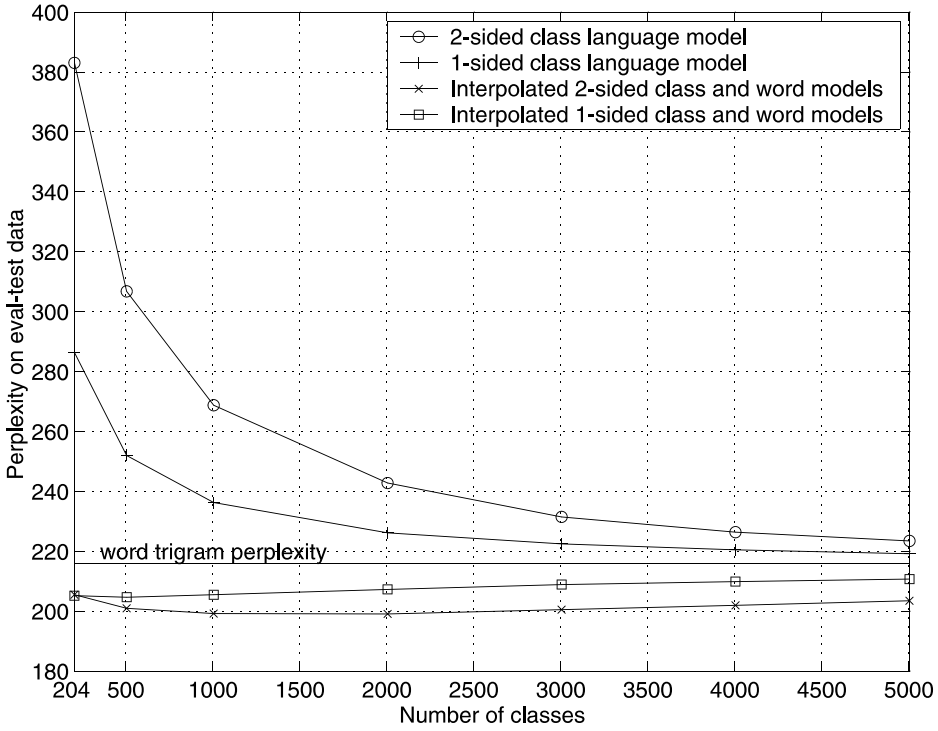

Fig. 1. Growth in vocabulary size against corpus size.

vocabulary, it cannot be recognised. Moreover, the wrong word that is hypothesised to have occurred in it place will affect the recognition of subsequent words since it will be used as the context for predicting the current word by the language model. It has been shown that, on average, for the Wall Street Journal Task, every OOV word that occurs in the test data, results in approximately 1.6 word-errors (Woodland et al., 1994).

We define a vocabulary of size N_V by taking the most frequent N_V words from the `training` set of each corpus. The OOV-rate is computed with respect to the `eval-test` set of each corpus. The results, displayed in Fig. 2, highlight the significant difference between the two languages which the size of the vocabulary has on the OOV-rate.

Currently the vocabulary size of a *large vocabulary speech recogniser* is around 65,000 (65k) words since this is close to the limit of the number of identifiers that can be represented in a computer by a two-byte integer. For the English corpus, such a vocabulary provides almost 99% coverage on the `eval-test` set. The most significant observation with regard to the Russian corpus, and one that inevitably dictates the course of subsequent work, is that the coverage of the appreciably large (65k) vocabulary is only 92.4%. Alternatively, the OOV-rate of 7.6% is seven times greater than for the English corpus 65k vocabulary.

From Fig. 2 we observe that as the size of the vocabulary is increased, the coverage on the English corpus increases almost ten times faster than the coverage on the Russian corpus. However, for both corpora and for the size of vocabularies of interest (>65k), each doubling of the vocabulary size approximately halves the OOV-rate. Note that there are almost always OOV

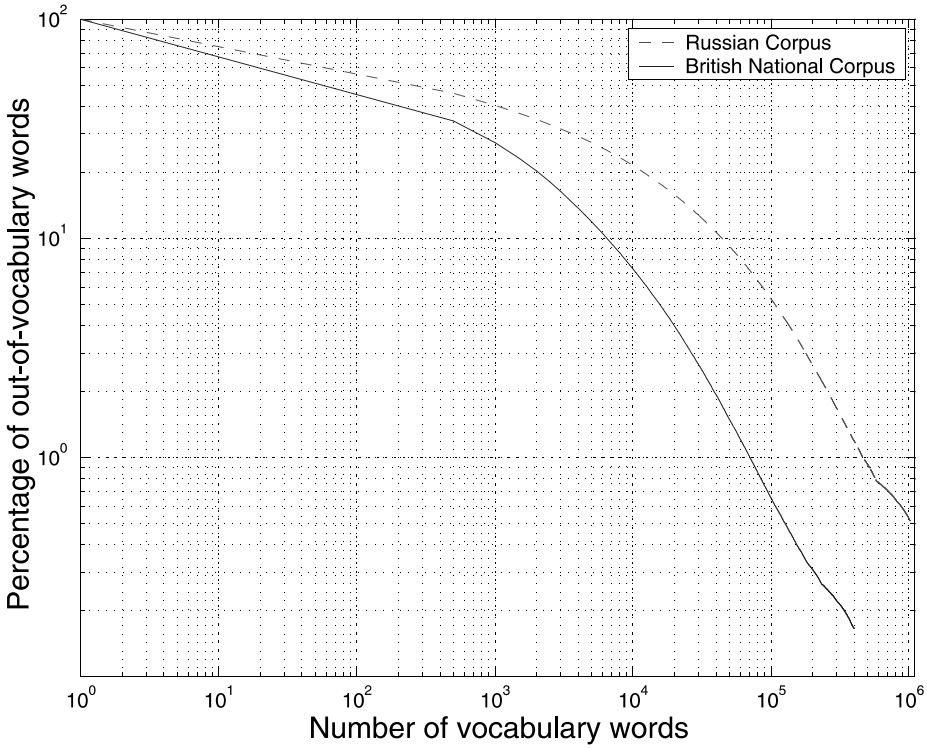

Fig. 2. Variation of OOV-rate against (log) vocabulary size.

words in some held-out text, even when the vocabulary is defined to contain all the words which occur in some much larger partition of the corpus.

All subsequent experiments will be performed with a vocabulary of the most frequent 65k words that occur in the `training` set for both the Russian and English corpora. In addition, experiments for a vocabulary of the most frequent 430,000 (430k) words in the `training` set will be used for the Russian corpus. This vocabulary size was chosen to provide an OOV-rate identical to that on the English corpus with a 65k vocabulary.

4. Word-based *N*-gram language modelling

Word trigram language models employing Good-Turing discounting and Katz back-off (Katz, 1987) were built on both corpora with all singleton N-grams ($N > 1$) discarded. A range of vocabulary sizes, up to the maximum in the `training` set were investigated to assess the variation in perplexity on the `eval-test` set. The perplexity of the word models increases as the vocabulary size is increased, since vocabulary words are chosen according to their frequency of occurrence in the corpus and increasing the vocabulary size means increasing the number of lower-frequency words in the vocabulary. The statistics of additional low-frequency words are unlikely to be as well estimated and will generally have a lower probability of occurring compared

Table 1
Perplexities of word trigram and 4-gram models with different cutoffs

	3-gram		4-gram	
Cutoffs (2g, 3g, 4g)	1, 1, _	0, 0, _	1, 1, 1	0, 0, 0
Russian 65k	413.3	387.4	398.9	385.5
Russian 430k	677.0	617.4	656.9	–
English 65k	216.1	208.4	200.6	199.1

to words in a smaller vocabulary. However, the increases in perplexity observed for English were found to be similar to those for Russian when examined over a comparable range of OOV-rates.

In Table 1, the perplexities on the `eval-test` data of Russian and English word trigram and 4-gram models with singleton events discarded $(1,1,\_/1,1,1)$ and retained $(0,0,\_/0,0,0)$ are given. The Russian 4-gram models[1] outperform the trigram models by around 3% compared to over 7% for English. This relatively small improvement for Russian may be a consequence of the sparsity of the Russian corpus, the effect of which becomes more significant as N is increased. It may also be speculated that the less constrained word-ordering in Russian means that there is little to be gained by increasing the context for making predictions, since sequences of longer length are as likely to appear in the future as hitherto unobserved permutations.

The results also show that, for Russian, retaining singleton N-grams is useful in reducing the perplexity for both trigram and 4-gram models and also for both vocabulary sizes. For the 65k trigram model the perplexity improvement is 6.3% while for the 430k trigram model it is 8.8%. Models in which all N-grams are retained, have significantly more parameters (five to six times more) than models which discard singleton N-grams. However, these are significant improvements and suggest that perhaps the Good-Turing discounting scheme in conjunction with backing-off is underestimating low-frequency events for Russian. The improvements when N-grams are retained correlate with an increase in trigram (4-gram) hits and further imply the usefulness of the singleton events in prediction. For the English models, on the other hand, when singleton events are retained there is a 3.6% improvement in the trigram model and a negligible improvement in the 4-gram model. This observation can be expected to hold true only for training and test data that are homogeneous.

5. Class-based N-gram language modelling

All class-based language models (hereafter referred to simply as class models) employ some component that uses word equivalence classes to capture dependencies in the training text. A deterministic word classification function (or class mapping function) of the form

$$C : w \rightarrow C(w) \tag{1}$$

assigns each word to one class only, hence the class mapping function is many-to-one. Word classes are typically groups of words which are deemed to be similar in some way. If linguistic

[1] The 430k Russian 4-gram with all events retained, $(0,0,0)$, could not be built due to memory limitations.

parts of speech are used to define the classes, all nouns might be grouped together in one of the classes, for example, and all adjectives in another. Alternatively, some statistical criterion of similarity can be used to determine the word classes. The latter is the focus of the class-based work in this paper.

Having determined C, many different forms of class language model can be formulated by making various approximations and assumptions on the dependence between words and classes, for example

$$P(w_i|w_{i-N+1},\ldots,w_{i-2},C(w_{i-1})), \tag{2}$$

$$P_0(w_i|C(w_i))\cdot P_1(C(w_i)|C(w_{i-N+1}),\ldots,C(w_{i-1})), \tag{3}$$

$$P(w_i|C(w_{i-N+1}),\ldots,C(w_{i-1})). \tag{4}$$

In this paper, we will consider only the last two class models, which we will refer to as the two-sided and one-sided class models, respectively. The two-sided class model is similar to the one-sided class model but makes the assumption that the probability of the current word is independent of the class of the previous word, if the class of the current word is known. The one-sided class bigram model can be thought of as a variation on the word N-gram model in which the word histories are tied so that the model parameters are more robustly estimated.

5.1. Two-sided class model

Ney refers to the model given by Eq. (3) as the two-sided symmetric class model (Ney et al., 1994) since the same word classification function C is used to map both the current word and the predecessor words. The model comprises two independent probability distributions: a unigram class membership component $P_0(\cdot)$ and a class N-gram component $P_1(\cdot)$ which is used to predict the current word's class from its predecessor word classes.

In this paper, C is determined by optimising the log-likelihood of a bigram class model LL_{bi} on the training data, using the relative frequency estimates for the component probabilities

$$LL_{\mathrm{bi}}(C)=\sum_{i=1}^{N_{\mathrm{W}}}\log\frac{N(w_i)}{N(C(w_i))}\cdot\frac{N(C(w_{i-1}),C(w_i))}{N(C(w_{i-1}))}, \tag{5}$$

where $N(\cdot)$ is the count of the event inside the brackets and N_{W} is the total number of words in the training data. By grouping together similar terms and noting that the classification function has no effect on terms containing only $N(w_i)$ we obtain the following optimisation function:

$$LL_{\mathrm{bi}}(C)=\sum_{i=1}^{N_{\mathrm{C}}}\sum_{j=1}^{N_{\mathrm{C}}}N(c_i,c_j)\cdot\log N(c_i,c_j)-2\cdot\sum_{i=1}^{N_{\mathrm{C}}}N(c_i)\cdot\log N(c_i), \tag{6}$$

where N_{C} is the number of word equivalence classes and N_{V} was defined in Section 3.5. Once C has been determined, it is a relatively simple matter to construct the class model. Although most algorithmic methods determine the classification function using a bigram class model, the assumption is often made that the classification is suitable for mapping all N words in the N-gram. There are far fewer free parameters to estimate in a class N-gram model than in a word N-gram

model since in general $N_C \ll N_V$; there are $(N_C \cdot (N_C - 1) + N_V - N_C)$ free parameters to estimate for a class bigram model compared to $N_V \cdot (N_V - 1)$ for a word bigram model.

5.2. One-sided class model

The two-sided symmetric class model combines what may be thought of as separate state transition and state emission distributions, with the same class mapping function for both the current and predecessor words. Here we consider the one-sided class N-gram model given by Eq. (4) in which a state mapping is used for the predecessor words only, and the current word that is being predicted is not mapped at all. The state mapping is an $(N-1)$-tuple of word classes, so the probability of the current word is directly conditioned on the $(N-1)$ classes of the predecessor words.

The optimisation criterion is defined to be the log-likelihood of the training text using a one-sided class bigram model and maximum likelihood estimates of the probabilities

$$LL_{\text{bi}}(C) = \sum_{i=1}^{N_W} \log P(w_i|C(w_{i-1})) = \sum_{j=1}^{N_C} \sum_{i=1}^{N_V} N(c_j, w_i) \cdot \log \frac{N(c_j, w_i)}{N(c_j)}, \tag{7}$$

which can be further simplified for implementation to

$$LL_{\text{bi}}(C) = \sum_{j=1}^{N_C} \sum_{i=1}^{N_V} N(c_j, w_i) \cdot \log N(c_j, w_i) - \sum_{j=1}^{N_C} N(c_j) \cdot \log N(c_j). \tag{8}$$

In the one-sided class bigram model there are $N_C \cdot (N_V - 1)$ free parameters to estimate.

5.3. The clustering operation

The exchange algorithm (Duda and Hart, 1973) is used to determine the class mapping function for the two-sided and one-sided class models by maximising the log-likelihood on the training data of the two-sided, symmetric class bigram model given by Eq. (3) or of the one-sided class bigram model given by Eq. (4), respectively. The algorithm itself has been extensively discussed in the literature and an analysis of the update equations and efficient implementation details are given for the two-sided model in Martin et al. (1998), for example. In order not to repeat that analysis here, only the details which are considered relevant to the experiments will be given where appropriate.

5.3.1. Update equations for one-sided clustering

Count updates for the one-sided model can be performed in a similar manner to those described for the two-sided model in Martin et al. (1998). Only the contribution of those counts that are affected by the movement of w_i from class c_j to class c_k need to be updated, i.e., only those stored bigram counts in which w_i appears. The count update equations are as follows:

$$\forall w : N(c_j, w) = N(c_j, w) - N(w_i, w), \tag{9}$$

$$\forall w : N(c_k, w) = N(c_k, w) + N(w_i, w), \tag{10}$$

$$N(c_j) = N(c_j) - N(w_i), \tag{11}$$

$$N(c_k) = N(c_k) + N(w_i), \tag{12}$$

where, for example,

$$N(c_k, w) = \sum_{\forall i: w_i \in c_k} N(w_i, w). \tag{13}$$

There is no explicit search involved in computing the count updates since only those counts for words which follow w_i in the training data need to be changed and these can be indexed directly. The contribution to the log-likelihood of w_i being in class c_j need only be computed once. Thereafter, the contribution to the log-likelihood of w_i being in all remaining classes c_k can be computed.

For the two-sided model when w_i is moved, the class bigram counts $N(c_k, c)$ and $N(c, c_k)$ for all c are affected. However, for the one-sided model only $N(c_k, w)$ is affected, hence the change in log-likelihood can effectively be computed for all classes simultaneously. Word w_i is then moved to the class c_k, for which the increase in log-likelihood is the greatest.

5.3.2. Computational complexity of two-sided clustering

The update equations presented in Martin et al. (1998) facilitate the computation of the optimisation criterion in $\mathcal{O}(N_C)$ time each time a word is moved from one class to another class. Since, for each of the I iterations, each word in the vocabulary must be moved (tentatively) from its original class to all possible destination classes, there is a $\mathcal{O}(I \cdot N_V \cdot N_C^2)$ complexity to the algorithm which dominates due to the log operations in the innermost loop where the change in log-likelihood is computed. However, for small numbers of classes, the size of the training data N_W may also become a dominant factor. This manifests itself as the number of unique bigrams (containing only vocabulary words) in the training data B^2 (where $B \ll N_W$ in general) which is factored into the count generation procedure. The complexity of the algorithm can therefore be shown to be

$$\mathcal{O}(I \cdot (2 \cdot B + 2 \cdot N_V \cdot N_C^2)), \tag{14}$$

where the $2 \cdot B$ factor originates from the generation of counts $N(w_i, c)$ and $N(c, w_i)$ and because the implementation does not involve any search in looking up the necessary bigram counts. In addition, by only considering the average number of predecessor word classes N_C^{pre} and successor word classes N_C^{suc}, for which $N(w_i, c) \neq 0$ and $N(c, w_i) \neq 0$ the complexity can be reduced still further to

$$\mathcal{O}(I \cdot (2 \cdot B + N_V \cdot N_C \cdot (N_C^{pre} + N_C^{suc}))). \tag{15}$$

This results in a significant reduction in the complexity of the algorithm which is consequently made to be "more than linear but far less than quadratic [in the number of classes]" (Martin et al., 1998).

[2] The number of distinct bigrams B in the training data is affected by the size of the training data N_W and the size of the vocabulary N_V. As an example of the absolute values that are involved, for the 65k Russian vocabulary $B \approx 12 \times 10^6$ and for the 430k vocabulary $B \approx 20 \times 10^6$.

5.3.3. Computational complexity for one-sided clustering

Each iteration I of the algorithm involves finding the locally optimal class for the N_V vocabulary words by moving each word in turn to each of the N_C classes. When word w_i is moved to a tentative destination class, $((B/N_V)+1)$ count updates involving a log operation must be performed on average, where (B/N_V) is the average number of distinct bigrams per word, i.e., all $N(w_i, w)$ in which w_i appears. The complexity of the algorithm is therefore linear in the number of classes and linear in the number of words in the vocabulary:

$$\mathcal{O}\left(I \cdot N_V \cdot N_C \cdot \left(\frac{B}{N_V}\right)\right). \tag{16}$$

Compared to Eq. (15) this complexity highlights a significant advantage of this algorithm over that for the two-sided model, specifically because the algorithm no longer scales quadratically in the number of classes and since $(B/N_V) \ll N_C^{pre} + N_C^{suc}$.

5.3.4. Efficient code implementation

Implementation of each algorithm depends largely on how the bigram counts are to be stored. For the two-sided clustering algorithm the bigram counts can easily be accommodated in a $N_C \times N_C$ array for the range of values of N_C that we investigate. For the one-sided clustering algorithm, the storage of the bigram counts would require a $N_V \times N_C$ array which is prohibitive for all but small values of N_V and N_C. We will refer to the latter implementation as version one of the one-sided clustering algorithm. Version two of the algorithm was implemented using linked-lists. This removes the necessity for any repetitive search operations during count generation and allows efficient storage of the bigram counts $N(c, w)$.

5.4. Experimental procedure

The experimental procedure employed for obtaining the word classes and subsequently building the class trigram models was identical for both corpora so as to allow as comprehensive and legitimate a comparison as possible. Each algorithm was initialised by assigning the most frequent $(N_C - 1)$ vocabulary words each to their own unique class and all remaining vocabulary words were placed in the N_Cth class. Words were clustered into 204, 504, 1004, 2004, 3004, 4004, and 5004 classes. The extra four classes relate to the four special symbols: ⟨s⟩, ⟨/s⟩, ⟨NUMBER⟩, and ⟨UNK⟩ (sentence-begin, sentence-end, number, and unknown-word symbols) which were each placed in their own unique classes and could not be moved from these classes nor could other words be moved to them during clustering. The vocabulary size was fixed to be the most frequent 65k words (also 430k words for the Russian experiments) as used in the word N-gram experiments in Section 4. All vocabulary words, other than the four special symbols, were considered for classification and were moved in order of decreasing frequency of occurrence in the training data. Two iterations through the vocabulary were performed in every case except for two-sided clustering of the Russian 430k vocabulary where N_C was limited to 2004 classes and only one iteration was executed due to the excessive amount of computation time required. The final classification function obtained was then used to construct Katz back-off class trigram models built and smoothed in an identical manner to the word N-gram models in Section 4. Following the observations made in the same section, it was considered more important that the class models did

not contain singleton N-gram events that had been discarded from the word model since these were considered more likely to contribute to any differences observed in model performance. As a consequence each class model contains a different number of explicitly stored unigram, bigram and trigram events.

5.5. Clustering times

A comparison of the clustering times per iteration between the one-sided and two-sided models' clustering algorithms is plotted in Fig. 3 for the Russian 65k and 430k vocabularies. The clustering times for the English 65k vocabulary, though not shown, are similar to those for the Russian 65k vocabulary. The clustering operation was performed on a 300 MHz Ultra 2 Sun Sparc workstation. The two points shown by circles in Fig. 3 are for version one of the one-sided clustering algorithm in which statically allocated arrays are used (see Section 5.3.4).

It is clear from Fig. 3 that for the 65k vocabulary, the one-sided clustering is much faster than two-sided clustering for $N_C \geqslant 504$ and for the 430k vocabulary this is also true when $N_C \geqslant 1004$. Moreover, using version one, the faster implementation of the one-sided algorithm, makes one-sided clustering faster for the values of $N_C(\leqslant 504)$ over which it was practical to use it, at the expense of vastly increased memory requirements. The almost quadratic increase in clustering

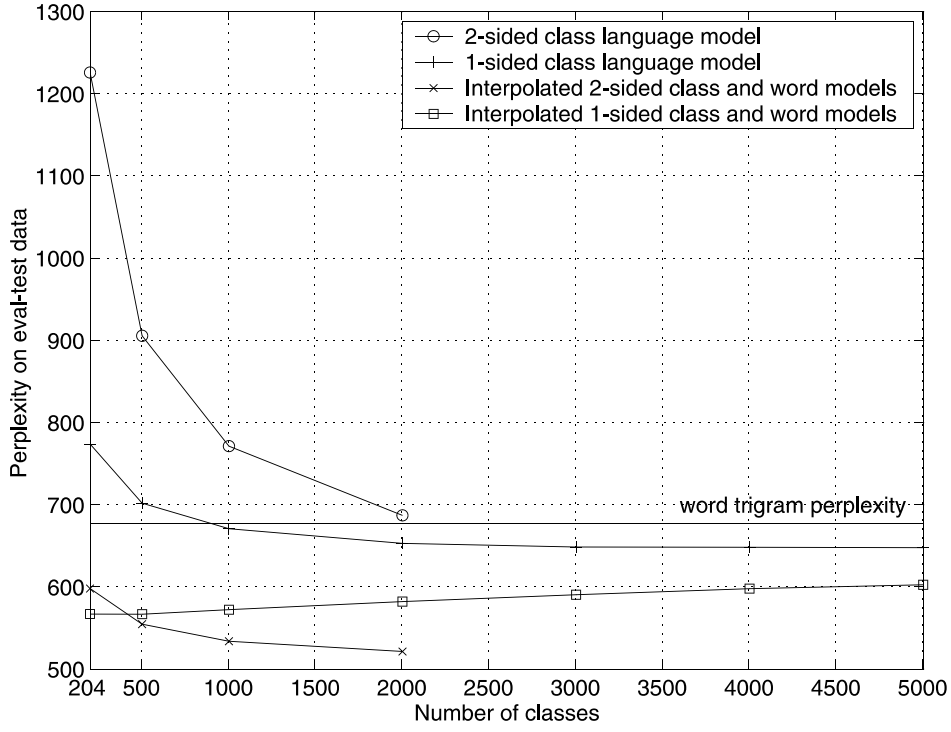

Fig. 3. Clustering times per iteration for one-sided and two-sided models with both a 65k and 430k Russian vocabulary on a 300 MHz Ultra 2 Sun Sparc workstation (version one of one-sided clustering algorithm is marked using circles).

time with the number of classes for the two-sided model is very clear from the figure as is the approximately linear increase in clustering time for one-sided clustering. This establishes the major advantage of the one-sided clustering algorithm over the two-sided clustering algorithm. Moreover, when a large vocabulary is used the excessive time required by the two-sided algorithm makes the one-sided algorithm the only practical choice between these two.

5.6. Perplexity results

The perplexities of the stand-alone class trigram models and of the interpolated word trigram and class trigram models computed on the `eval-test` set of each corpus are shown in Figs. 4–6. The interpolation weights were chosen so as to optimise the perplexity on the appropriate `dev-test` portion of each corpus using a tool that uses the E-M algorithm (Rosenfeld, 1994) to perform the optimisation. Figs. 4 and 5 show perplexity results on Russian for the 65k and 430k vocabularies respectively. Fig. 6 shows the perplexity results for the 65k English vocabulary.

It is clear that the overall trends across both languages, and across both vocabulary sizes for Russian, are more or less identical. However, an interesting difference between the two languages is that for both Russian vocabularies the one-sided class models with $N_C > 1004$ have a lower perplexity than the word model, whereas for English this does not occur in the range of classes that were investigated. For the 430k Russian vocabulary the one-sided model has a perplexity up

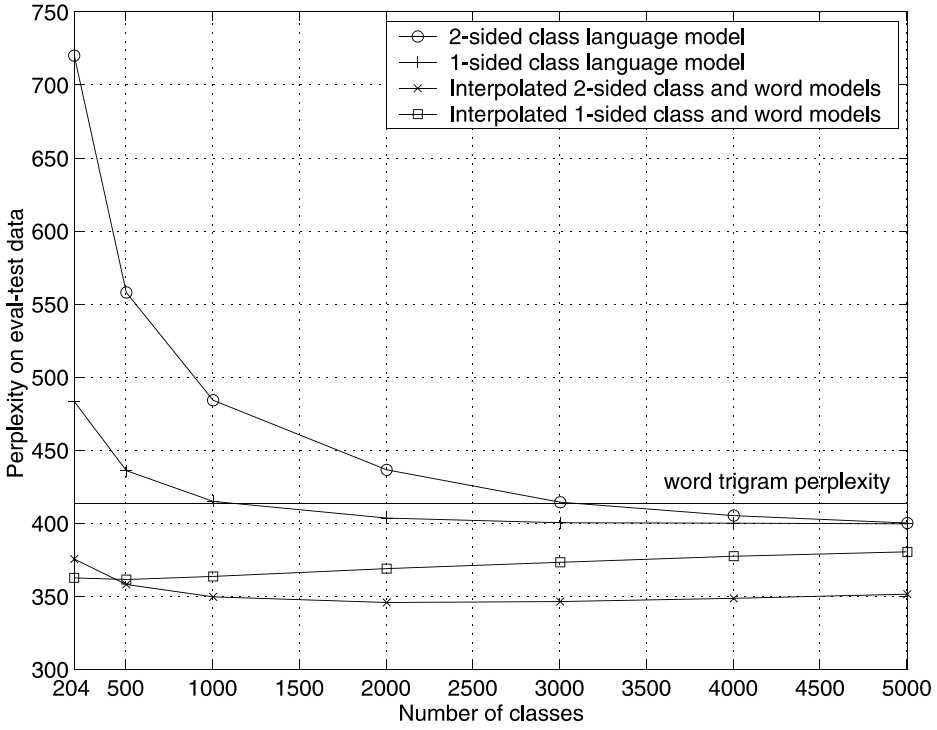

Fig. 4. Russian (65k): perplexity results for stand-alone class and interpolated word/class models on `eval-test` data.

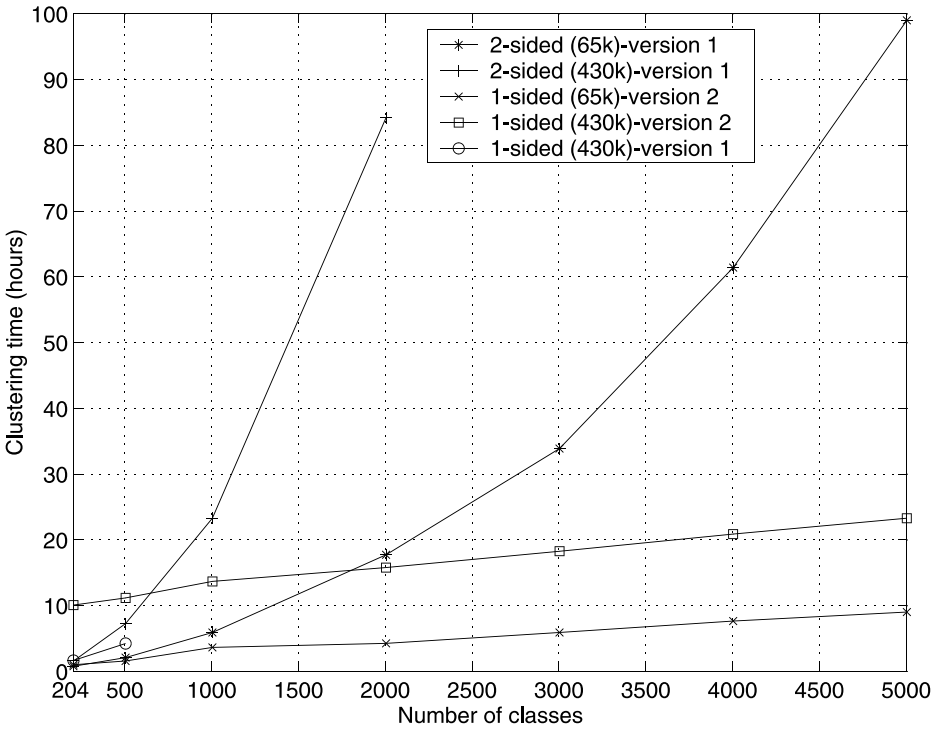

Fig. 5. Russian (430k): perplexity results for stand-alone class and interpolated word/class models on `eval-test` data. (Only one iteration of two-sided clustering algorithm was executed to obtain the two-sided results.).

to 4.3% lower than the word model. Moreover, all the one-sided class models had fewer parameters than the word model.

When the class models are linearly interpolated with the word model, the weight assigned to the class model was found to increase as N_C increased. However, the word model was always given a higher weighting even when interpolated with a class model that had a lower perplexity than the word model. For both languages with the 65k vocabularies the interpolated models exhibit a minimum perplexity when 2004 classes are used in the two-sided model and when 504 classes are used in the one-sided model. We may expect that these minimum values are related to the quantity of training data used for building the models and the size of the vocabulary. In addition, since the minima for each type of class model occur with different numbers of classes this also increases the relative difference in clustering times required to build the best of each class model. Again, this favours one-sided clustering over two-sided clustering. We cannot say anything for certain about the 430k Russian experiments since obtaining more data points was too time consuming. Nonetheless it is clear that the reduction in perplexity is greatest with the interpolated word and two-sided class model of up to 7.9% for the English vocabulary, and up to 23.0% with the 430k Russian vocabulary. The reduction obtained with the interpolated word and one-sided class model is up to 5.3% for English and up to 16.2% for the 430k Russian vocabulary. Linear interpolation has effectively combined the generalisation strength of the class model with the specificity strength of the word model.

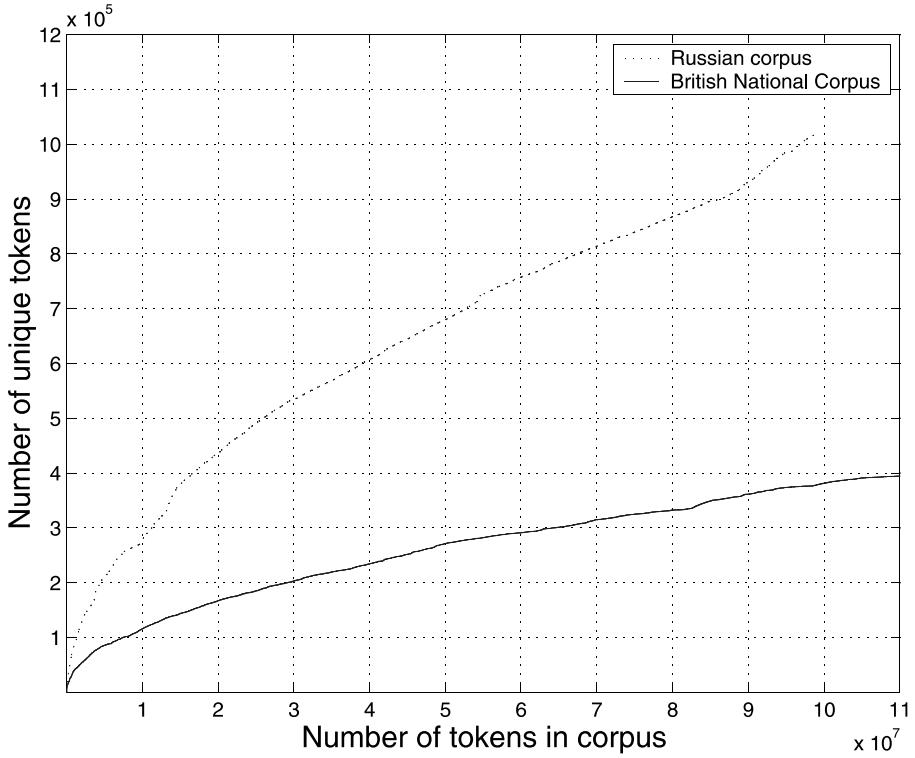

Fig. 6. English (65k): perplexity results for stand-alone class and interpolated word/class models on `eval-test` data.

For a given value of N_C the one-sided class model tends to have more parameters than the two-sided model. This is directly related to the types of dependencies that the two types of class models capture. In general, it was observed that one-sided and two-sided class models with similar numbers of parameters had the same perplexity when they were used alone. The different types of dependencies in each model also explains the different optimal values of N_C for the interpolated word and class models. The two-sided class model captures more general dependencies than the one-sided model and a higher value of N_C is required in the interpolated model for the generalisation ability of the class model to best complement the specificity of the word model. The one-sided model captures dependencies that lie somewhere between the specific word sequences captured by the word model and the more general class sequences of the two-sided class model. The value of N_C affects the tradeoff between the ability to generalise to unseen word sequences and the accuracy with which words are predicted.

Examples of the contents of ten randomly chosen classes from a two-sided class model on English are given in Table 2. An examination of the contents of the classes from different models showed that words with a clear semantic or syntactic relationship had often been clustered together. This was true for both the one-sided and two-sided classifications. For Russian these relationships appeared even more obvious, for example, there were many classes in which most of the inflected forms of a particular word had been grouped together.

Table 2
All, or up to the 10 most frequent, words from 10 randomly chosen classes of the 1004-class two-sided English model

Class 1	Class 2	Class 3	Class 4	Class 5
				CONCERNED
				WRONG
				HAPPENING
EVEN	I'D	HIMSELF		RAINING
EQUIVALENTLY	YOU'D	MINDEDLY	EIGHT	BERSERK
	WE'D	THEMSELF		SNOWING
				GROUNDLESS
				NODED
				AGOG
				NEEDFUL
Class 6	Class 7	Class 8	Class 9	Class 10
	AVAILABLE	SOCIETY	RELATIONSHIP	GONE
	PAYABLE	CULTURE	RELATIONSHIPS	GROWN
	PRICED	POLITICS	CONFLICT	BEGUN
TO	REDUNDANT	LITERATURE	LINK	FALLEN
TER	TRANSMITTED	DEMOCRACY	CONVERSATION	SPOKEN
TAE	UNDERWAY	RELIGION	CONNECTION	RISEN
	UNAVAILABLE	CONSCIOUSNESS	LINKS	ARISEN
	REFERENCED	IDEOLOGY	COMPARISON	SPRUNG
	RECOVERABLE	CAPITALISM	PARTNERSHIP	LAIN
	OBTAINABLE	CHRISTIANITY	TENSION	SHRUNK

6. Word recognition experiments

Combinations of words and classes have often given improvements over using only word models on broadcast news tasks. In this section, we examine the difference in performance between the one-sided and two-sided class models when they are interpolated with a word model on a broadcast news task.

The 1997 DARPA HUB4 broadcast news evaluation was chosen for the experiments and we perform lattice rescoring experiments on lattices generated using the 1997 HTK broadcast news transcription system described in Woodland et al. (1998). The language model training data comprised 132 million words of the LDC broadcast news texts, the transcriptions of the 1997 broadcast news training data (added twice) and the 1995 Marketplace transcriptions. A word trigram model was built using the same vocabulary that was used to generate the original lattices. This baseline word trigram employed Katz back-off with Good-Turing discounting and had singleton bigrams and trigrams removed to produce a model containing around 16.5 million parameters. The optimal number of classes and the interpolation weights between the word and class models were optimised on the development lattices. The number of classes was varied in increments of 100 between 100 and 1500 classes and the interpolation weights evaluated in increments of 0.1. The optimal number of classes for the two-sided model was found to be 1000 with interpolation weights of 0.7 (word) and 0.3 (class). For the one-sided model the optimal number of classes was 400 with interpolation weights 0.6 (word) and 0.4 (class). The perplexity of the models on the reference transcription and the word error rate results are given in Table 3.

Table 3
Perplexity and word error rate on evaluation data of the optimised, interpolated word and class models, and the baseline word trigram model

Model	PP_{ref}	%*WER*	% Improvement
Interpolated two-sided 1000	161.6	17.8	2.2
Interpolated one-sided 400	162.4	17.8	2.2
Baseline word trigram	171.4	18.2	–

Both model combinations give an improvement in performance over the baseline word trigram model, and each improvement is statistically significant at the 99% level using the NIST Matched Pair Sentence Segment test. Also, interpolating a word 4-gram model (with bigram, trigram and 4-gram cutoffs of 1,3,3) with a class 4-gram model (with the same cutoffs) reduced the word 4-gram baseline result of 17.6–17.1% for the interpolated word and two-sided model, and to 17.2% for the interpolated word and one-sided model. Both these improvements were also found to be statistically significant at the 99% confidence level using the same test.

7. Discussion

It is clear from the perplexity results that have been presented for Russian that combined class and word-based language modelling can produce significant improvements in performance. For a language like Russian where higher order word N-gram models do not significantly improve performance, combinations of word and class models appear to offer an appealing solution. Although the improvements in perplexity were shown to be generally less for English, they still translated into significant reductions in word error rate on an English language broadcast news task.

The advantage of using the one-sided class model has been clearly demonstrated for a situation where a very large vocabulary is necessary. Automatic classifications can be obtained in significantly less time than for the two-sided class model with little loss in performance. Also, although it has not been explicitly investigated here, clustering can be used as a pruning technique for obtaining smaller and/or more robust stand-alone language models. This was demonstrated with the perplexity results presented in Section 5.6. For such situations a larger number of classes is generally required hence the scaling properties of the algorithm for the one-sided class model recommend its use.

An investigation of grammar-based approaches, and language modelling with non-local dependencies, to tackle the outstanding issues in Russian is left for future work. We believe, nonetheless, that the models and results presented in this paper provide a good reference point for such future experiments.

8. Conclusion

The first observation that was made regarding the differences between Russian and English was that to achieve the same coverage as a 65k vocabulary on English, a vocabulary approximately

seven times larger is required for Russian. It was shown that the use of higher order N-grams was generally not as effective for modelling Russian as it was for English. This prompted the examination of combinations of word and class models since, in the literature, such combinations had been shown to be effective at reducing the perplexity on data in several languages. Automatic classifications of words into classes had also been shown to be superior to linguistically derived classifications. Two types of class model were investigated in the paper: a two-sided class model and a novel, one-sided class model. The experiments in this paper represent the first investigation of the characteristics of the one-sided class model. Results of perplexity experiments using combinations of word and class models showed that the combination with the two-sided model produced greater reductions in perplexity than the combination with the one-sided model for both languages and vocabulary sizes considered. However, the clustering algorithm used to obtain classifications for the two-sided class model was shown to be extremely slow for the very large Russian vocabulary. The algorithm for the one-sided model, on the other hand, was shown to be much faster. Finally, the results of lattice rescoring experiments on an English language broadcast news task showed that word error rate reductions were both significant and identical for both combinations of the word model with either the one-sided or two-sided class model.

References

Burnard, L., 1995. Users Reference Guide for the British National Corpus. Oxford University Computing Services.

Duda, R.O., Hart, P.E., 1973. In: Pattern Classification and Scene Analysis. Wiley, New York, pp. 227–228.

Katz, S.M., 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustic, Speech, and Signal Processing 35 (3), 400–401.

Martin, S., Liermann, J., Ney, H., 1998. Algorithms for bigram and trigram word clustering. Speech Communication 24, 19–37.

Ney, H., Essen, U., Kneser, R., 1994. On structuring probabilistic dependences in stochastic language modelling. Computer Speech and Language 8, 1–38.

Rosenfeld, R., 1994. Adaptive statistical language modelling: a maximum entropy approach. Ph.D. thesis, School of Computer Science, Carnegie Mellon University. Technical Report CMU-CS-94-138.

Whittaker, E.W.D., 2000. Statistical language modelling for automatic speech recognition of Russian and English. Ph.D. thesis, Cambridge University.

Woodland, P.C., Hain, T., Johnson, S.E., Niesler, T.R., Tuerk, A., Whittaker, E.W.D., Young, S.J., 1998. The 1997 HTK broadcast news transcription system. In: Proceedings of the 1998 DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, pp. 41–48.

Woodland, P.C., Odell, J.J., Valtchev, V., Young, S.J., 1994. Large vocabulary continuous speech recognition using HTK. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia.