VLC'00, pp.80-86, 2000.10.

Nonlocal Language Modeling based on Context Co-occurrence Vectors

Sadao Kurohashi and Manabu Ori
Graduate School of Informatics
Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606-8501 Japan
kuro@i.kyoto-u.ac.jp, ori@pine.kuee.kyoto-u.ac.jp

Abstract

This paper presents a novel nonlocal language model which utilizes contextual information. A reduced vector space model calculated from co-occurrences of word pairs provides word co-occurrence vectors. The sum of word co-occurrence vectors represents the context of a document, and the cosine similarity between the context vector and the word co-occurrence vectors represents the long-distance lexical dependencies. Experiments on the Mainichi Newspaper corpus show significant improvement in perplexity (5.0% overall and 27.2% on target vocabulary)

1 Introduction

Human pattern recognition rarely handles isolated or independent objects. We recognize objects in various spatiotemporal circumstances such as an object in a scene, a word in an utterance. These circumstances work as conditions, eliminating ambiguities and enabling robust recognition. The most challenging topics in machine pattern recognition are in what representation and to what extent those circumstances are utilized.

In language processing, a context—that is, a portion of the utterance or the text before the object—is an important circumstance. One way of representing a context is statistical language models which provide a word sequence probability, $P(w_1^n)$, where w_i^j denotes the sequence $w_i \ldots w_j$. In other words, they provide the conditional probability of a word given with the previous word sequence, $P(w_i|w_1^{i-1})$, which shows the prediction of a word in a given context.

The most common language models used nowadays are N-gram models based on a $(N-1)$-th order Markov process: event predictions depend on at most $(N-1)$ previous events. Therefore, they offer the following approximation:

$$P(w_i|w_1^{i-1}) \approx P(w_i|w_{i-N+1}^{i-1}) \qquad (1)$$

A common value for N is 2 (bigram language model) or 3 (trigram language model); only a short local context of one or two words is considered.

Even such a local context is effective in some cases. For example, in Japanese, after the word *kokumu* 'state affairs', words such as *daijin* 'minister' and *shou* 'department' likely follow; *kaijin* 'monster' and *shou* 'prize' do not. After *dake de* 'only at', you can often find *wa* (topic-marker), but you hardly find *ga* (nominative-marker) or *wo* (accusative-marker). These examples show behaviors of compound nouns and function word sequences are well handled by bigram and trigram models. These models are exploited in several applications such as speech recognition, optical character recognition and morphological analysis.

Local language models, however, cannot predict much in some cases. For instance, the word probability distribution after *de wa* 'at (topic-marker)' is very flat. However, even if the probability distribution is flat in local language models, the probability of *daijin* 'minister' and *kaijin* 'monster' must be very different in documents concerning politics. Bigram and trigram models are obviously powerless to such kind of nonlocal, long-distance lexical dependencies.

This paper presents a nonlocal language model. The important information concerning long-distance lexical dependencies is the word co-occurrence information. For example, words such as politics, government, administration, department, tend to co-occur with *daijin* 'minister'. It is easy to measure co-occurrences of word pairs from a training corpus, but utilizing them as a representation of context is the problem. We present a vector

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
w_1	1	0	1	0	1	0	1	0
w_2	1	0	1	1	0	0	0	0
w_3	0	1	0	0	1	1	0	1
w_4	1	1	1	0	0	0	0	0
w_5	0	0	0	0	1	0	1	0
w_6	0	0	0	0	1	0	0	1

Figure 1: Word-document co-occurrence matrix.

	w_1	w_2	w_3	w_4	w_5	w_6
w_1	4	2	1	2	2	1
w_2		3	0	2	0	0
w_3			4	1	1	2
w_4				3	0	0
w_5					2	1
w_6						2

Figure 2: Word-word co-occurrence matrix.

representation of word co-occurrence information, and show that the context can be represented as a sum of word co-occurrence vectors in a document and it is incorporated in a non-local language model.

2 Word Co-occurrence Vector

2.1 Word-Document Co-occurrence Matrix

Word co-occurrences are directly represented in a matrix whose rows correspond to words and whose columns correspond to documents (e.g. a newspaper article). The element of the matrix is 1 if the word of the row appears in the document of the column (Figure 1). We call such a matrix a *word-document co-occurrence matrix*.

The row-vectors of a word-document co-occurrence matrix represent the co-occurrence information of words. If two words tend to appear in the same documents, that is, tend to co-occur, their row-vectors are similar, that is, they point in similar directions.

The more document is considered, the more reliable and realistic the co-occurrence information will be. Then, the row size of a word-document co-occurrence matrix may become very large. Since enormous amounts of online text are available these days, row size can become more than a million documents. Then, it is not practical to use a word-document co-occurrence matrix as it is. It is necessary to reduce row size and to simulate the tendency in the original matrix by a reduced matrix.

2.2 Reduction of Word-Document Co-occurrence Matrix

The aim of a word-document co-occurrence matrix is to measure co-occurrence of two words by the angle of the two row-vectors. In the reduction of a matrix, angles of two row-vectors in the original matrix should be maintained in the reduced matrix.

As such a matrix reduction, we utilized a learning method developed by HNC Software (Ilgen and Rushall, 1996). [1]

1. Not the word-document co-occurrence matrix is constructed from the learning corpus, but a word-word co-occurrence matrix. In this matrix, the rows and columns correspond to words and the i-th diagonal element denotes the number of documents in which the word w_i appears, $F(w_i)$. The i,j-th element denotes the number of documents in which both words w_i and w_j appear, $F(w_i, w_j)$ (Figure 2).

   The important information in a word-document co-occurrence matrix is the cosine of the angle of the row-vector of w_i and that of w_j, which can be calculated by the word-word co-occurrence matrix as follows:

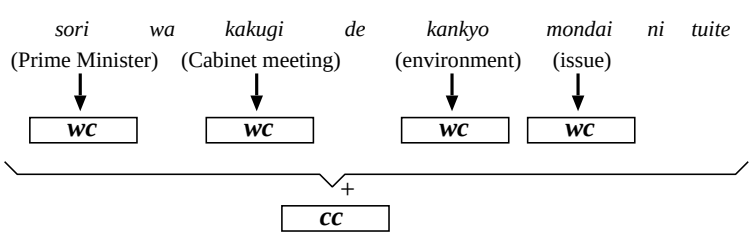
   $$\alpha_{ij} = \frac{F(w_i, w_j)}{\sqrt{F(w_i)}\sqrt{F(w_j)}} \qquad (2)$$

   This is because $\sqrt{F(w_i)}$ corresponds to the magnitude of the row-vector of w_i, and $F(w_i, w_j)$ corresponds to the dot product of the row-vector of w_i and that of w_j in the word-document co-occurrence matrix.

2. Given a reduced row size, a matrix is initialized as follows: matrix elements are chosen from a normal distribution randomly, then each row-vector is normalized to magnitude 1.0. The random unit row-vector of the word w_i is denoted as $\boldsymbol{wc}_i^{Rand}$.

   Random unit row-vectors in high dimensional floating point spaces have a

[1] The goal of HNC was the enhancement of text retrieval. The reduced word vectors were regarded as semantic representation of words and used to represent documents and queries.



w	$(cc \bullet wc)^2$	Pc
kaigi (conference)	0.237962	0.002702
senkyo (election)	0.150773	0.001712
yosan (budget)	0.128907	0.001463
daijin (minister)	0.018549	0.000211
yakyu (baseball)	0.004556	0.000052
kaijin (monster)	0.000002	0.000000
sugaku (mathematics)	0.000001	0.000000
....	....	....
TOTAL	88.079230	1.000000

Figure 3: An example of context co-occurrence probabilities.

property that is referred to a "quasi-orthogonality". That is, the expected value of the dot product between any pair of random row-vectors, $\boldsymbol{wc}_i^{Rand}$ and $\boldsymbol{wc}_j^{Rand}$, is approximately equal to zero (i.e. all vectors are approximately orthogonal).

3. The trained row-vector, $\boldsymbol{wc}_i$ is calculated as follows:

$$\boldsymbol{wc}_i = \boldsymbol{wc}_i^{Rand} + \eta \sum_j \alpha_{ij} \boldsymbol{wc}_j^{Rand} \tag{3}$$

$$\boldsymbol{wc}_i = \frac{\boldsymbol{wc}_i}{||\boldsymbol{wc}_i||} \tag{4}$$

α_{ij} corresponds to the degree of the co-occurrence of two words. By adding $\boldsymbol{wc}_j^{Rand}$ to $\boldsymbol{wc}_i^{Rand}$ depending on α_{ij}, the learning formula (3) achieves that two words that tend to co-occur will have trained vectors that point in similar directions. η is a design parameter chosen to optimize performance. The formula (4) is to normalize vectors to magnitude 1.0.

We call the trained row-vector $\boldsymbol{wc}_i$ of the word w_i a *word co-occurrence vector*.

The background of the above method is a stochastic gradient descent procedure for minimizing the cost function:

$$J = \frac{1}{2} \sum_{i,j} (\alpha_{ij} - \boldsymbol{wc}_i \cdot \boldsymbol{wc}_j)^2 \tag{5}$$

subject to the constraints $||\boldsymbol{wc}_i|| = 1$.

The procedure iterates the following calculation:

$$\begin{aligned} \boldsymbol{wc}_i^{new} &= \boldsymbol{wc}_i - \eta \frac{\partial J}{\partial \boldsymbol{wc}_i} \\ &= \boldsymbol{wc}_i + \eta \sum_j (\alpha_{ij} - \boldsymbol{wc}_i \cdot \boldsymbol{wc}_j) \boldsymbol{wc}_j \end{aligned} \tag{6}$$

$$\boldsymbol{wc}_i^{new} = \frac{\boldsymbol{wc}_i^{new}}{||\boldsymbol{wc}_i^{new}||} \tag{7}$$

The learning method by HNC is a rather simple approximation of the procedure, doing just one step of it. Note that $\boldsymbol{wc}_i \cdot \boldsymbol{wc}_j$ is approximately zero for the initialized random vectors.

3 Context Co-occurrence Vector

The next question is how to represent the context of a document based on word co-occurrence vectors. We propose a simple model which represents the context as the sum of the word co-occurrence vectors associated with content words in a document so far. It should be noted that the vector is normalized to unit length. We call the resulting vector a *context co-occurrence vector*.

Word co-occurrence vectors have the property that words which tend to co-occur have vectors that point in similar directions. Context co-occurrence vectors are expected to have the similar property. That is, if a word tends to appear in a given context, the word co-occurrence vector of the word and the context co-occurrence vector of the context will point in similar directions.

Such a context co-occurrence vector can be seen to predict the occurrence of words in a

$$P(w_i|w_1^{i-1}) = \begin{cases} P(C_c|w_1^{i-1}) \times P(w_i|w_1^{i-1}C_c) & \text{if } w_i \in C_c \\ P(C_f|w_1^{i-1}) \times P(w_i|w_1^{i-1}C_f) & \text{if } w_i \in C_f \end{cases}$$

where

$$\begin{aligned} P(C_c|w_1^{i-1}) &= \lambda_1 P(C_c) + \lambda_2 P(C_c|w_{i-1}) + \lambda_3 P(C_c|w_{i-2}w_{i-1}) \\ P(w_i|w_1^{i-1}C_c) &= \lambda_{c1} P(w_i|C_c) + \lambda_{c2} P(w_i|w_{i-1}C_c) + \lambda_{c3} P(w_i|w_{i-2}w_{i-1}C_c) \\ &\quad + \lambda_{cc} Pc(w_i|w_1^{i-1}C_c) \\ P(C_f|w_1^{i-1}) &= 1 - P(C_c|w_1^{i-1}) \\ P(w_i|w_1^{i-1}C_f) &= \lambda_{f1} P(w_i|C_f) + \lambda_{f2} P(w_i|w_{i-1}C_f) + \lambda_{f3} P(w_i|w_{i-2}w_{i-1}C_f) \end{aligned}$$

with

$$\lambda_1 + \lambda_2 + \lambda_3 = 1,\ \lambda_{c1} + \lambda_{c2} + \lambda_{c3} + \lambda_{cc} = 1,\ \lambda_{f1} + \lambda_{f2} + \lambda_{f3} = 1.$$

Figure 4: Context language model.

given context, and is utilized as a component of statistical language modeling, as shown in the next section.

4 Language Modeling using Context Co-occurrence Vector

4.1 Context Co-occurrence Probability

The dot product of a context co-occurrence vector and a word co-occurrence vector shows the degree of affinity of the context and the word. The probability of a content word based on such dot products, called a *context co-occurrence probability*, can be calculated as follows:

$$Pc(w_i|w_1^{i-1}C_c) = \frac{f(\boldsymbol{cc}_1^{i-1} \cdot \boldsymbol{wc}_i)}{\sum_{w_j \in C_c} f(\boldsymbol{cc}_1^{i-1} \cdot \boldsymbol{wc}_j)} \quad (8)$$

where $\boldsymbol{cc}_1^{i-1}$ denotes the context co-occurrence vector of the left context, $w_1 \ldots w_{i-1}$, and C_c denotes a content word class. $Pc(w_i|w_1^{i-1}C_c)$ means the conditional probability of w_i given that a content word follows w_1^{i-1}.

One choice for the function $f(x)$ is the identity. However, a linear contribution of dot products to the probability results in poorer estimates, since the differences of dot products of related words (tend to co-occur) and unrelated words are not so large. Experiments showed that x^2 or x^3 is a better estimate.

An example of context co-occurrence probabilities is shown in Figure 3.

4.2 Language Modeling using Context Co-occurrence Probability

Context co-occurrence probabilities can handle long-distance lexical dependencies while a standard trigram model can handle local contexts more clearly: in this way they complement each other. Therefore, language modeling of their linear interpolation is employed. Note that the linear interpolation of unigram, bigram and trigram models is simply referred to 'trigram model' in this paper.

The proposed language model, called a *context language model*, computes probabilities as shown in Figure 4. Since context co-occurrence probabilities are considered only for content words (C_c), probabilities are calculated separately for content words (C_c) and function words (C_f).

$P(C_c|w_1^{i-1})$ denotes the probability that a content word follows w_1^{i-1}, which is approximated by a trigram model. $P(w_i|w_1^{i-1}C_c)$ denotes the probability that w_i follows w_1^{i-1} given that a content word follows w_1^{i-1}, which is a linear interpolation of a standard trigram model and the context co-occurrence probabilities.

In the case of a function word, since the context co-occurrence probability is not considered, $P(w_i|w_1^{i-1}C_f)$ is just a standard trigram model.

λ's adapt using an EM re-estimation procedure on the held-out data.

Table 1: Perplexity results for the standard trigram model and the context language model.

	Language Model			Perplexity on the entire vocabulary	Perplexity on the target vocabulary
Standard Trigram Model				107.7	1930.2
Context Language Model					
	Vector size	η	$f(x)$		
	500	0.5	x^2	106.3 (-1.3%)	1663.8 (-13.8%)
	1000	0.3	x^2	102.7 (-4.7%)	1495.9 (-22.5%)
	1000	0.5	x	103.6 (-3.9%)	1496.1 (-22.5%)
*	1000	0.5	x^2	102.4 (-5.0%)	1406.2 (-27.2%)
	1000	0.5	x^3	102.4 (-5.0%)	1416.8 (-26.9%)
	1000	1.0	x^2	102.5 (-4.8%)	1430.3 (-25.9%)
	2000	0.5	x^2	102.4 (-5.0%)	1408.1 (-27.1%)
Standard Bigram Model				130.28	2719.67
Context Language Model					
	1000	0.5	x	125.06 (-4.0%)	2075.10 (-23.7%)
	1000	0.5	x^2	122.85 (-5.7%)	1933.68 (-28.9%)

bei ‘US’ *kabushiki* ‘stock’ *shijyo* ‘market’ *no* *kyutou* ‘sudden rise’ *wo* *haikei* ‘background’ *ni* *Wall-gai* ‘Wall Street’ *ga* *kakkyou* ‘activity’ *wo* *teishi* ‘show’ ,
wagayonoharu ‘prosperity’ *wo* *ouka* ‘enjoy’ *shite* ‘do’ *iru* . *shouken* ‘stock’ *kaisha* ‘company’ , *toushi* ‘investment’ *ginkou* ‘bank’ *wa* *1996* *nen* ‘year’ *ni*
haitte ‘enter’ *kara* *kako* ‘past’ *saikou* ‘maximum’ *eki* ‘profit’ *wo* *koushin* ‘renew’ . *’96* *nen* ‘year’ *no* *kabushiki* ‘stock’ *souba* ‘market’ *wa* *’95* *nen* ‘year’
ni *tsuzuki* ‘continue’ *kyushin* ‘rapid increase’ . *mata* *kabuka* ‘stock price’ *kyushin* ‘rapidly increase’ *wo* *haikei* ‘background’ *ni* *kigyou* ‘corporation’ *no*
shinkabu ‘new stock’ *hakkou* ‘issue’ *ga* *kako* ‘past’ *saikou* ‘maximum’ *to* ‘become’ *natta* .

Figure 5: Comparison of probabilities of content words by the trigram model and the context model. (Note that *wa, ga, wo, ni, to* and *no* are Japanese postpositions.)

4.3 Test Set Perplexity

By using the Mainichi Newspaper corpus (from 1991 to 1997, 440,000 articles), test set perplexities of a standard trigram/bigram model and the proposed context language model are compared. The articles of six years were used for the learning of word co-occurrence vectors, unigrams, bigrams and trigrams; the articles of half a year were used as a held-out data for EM re-estimation of λ's; the remaining articles (half a year) for computing test set perplexities.

Word co-occurrence vectors were computed for the top 50,000 frequent content words (excluding pronouns, numerals, temporal nouns, and light verbs) in the corpus, and unigram, bigram and trigram were computed for the top 60,000 frequent words.

The upper part of Table 1 shows the comparison results of the standard trigram model and the context language model. For the best parameters (marked by *), the overall perplexity decreased 5.0% and the perplexity on target vocabulary (50,000 content words) decreased 27.2% relative to the standard trigram model. For the best parameters, λ's were adapted as follows:

$$\lambda_1 = 0.08, \lambda_2 = 0.50, \lambda_3 = 0.42$$
$$\lambda_{c1} = 0.03, \lambda_{c2} = 0.50, \lambda_{c3} = 0.30, \lambda_{cc} = 0.17$$
$$\lambda_{f1} = 0.06, \lambda_{f2} = 0.57, \lambda_{f3} = 0.37$$

As for parameter settings, note that performance is decreased by using shorter word co-occurrence vector size. The variation of η does not change the performance so much.

$f(x) = x^2$ and $f(x) = x^3$ are almost the same, better than $f(x) = x$.

The lower part of Table 1 shows the comparison results of the standard bigram model and the context language model. Here, the context language model is based on the bigram model, that is, the terms concerning trigram in Figure 4 were eliminated. The result was similar, but the perplexity decreased a bit more; 5.7% overall and 28.9% on target vocabulary.

Figure 5 shows a test article in which the probabilities of content words by the trigram model and the context model are compared. If that by the context model is bigger (i.e. the context model predicts better), the word is boxed; if not, the word is underlined.

The figure shows that the context model usually performs better after a function word, where the trigram model usually has little prediction. On the other hand, the trigram model performs better after a content word (i.e. in a compound noun) because a clear prediction by the trigram model is reduced by paying attention to the relatively vague context co-occurrence probability (λ_{cc} is 0.17).

The proposed model is a constant interpolation of a trigram model and the context co-occurrence probabilities. More adaptive interpolation depending on the N-gram probability distribution may improve the performance.

5 Related Work

Cache language models (Kuhn and de Mori, 1990) boost the probability of the words already seen in the history.

Trigger models (Lau et al., 1993), even more general, try to capture the co-occurrences between words. While the basic idea of our model is similar to trigger models, they handle co-occurrences of word pairs independently and do not use a representation of the whole context. This omission is also done in applications such as word sense disambiguation (Yarowsky, 1994; FUNG et al., 1999).

Our model is the most related to Coccaro and Jurafsky (1998), in that a reduced vector space approach was taken and context is represented by the accumulation of word co-occurrence vectors. Their model was reported to decrease the test set perplexity by 12%, compared to the bigram model. The major differences are:

1. SVD (Singular Value Decomposition) was used to reduce the matrix which is common in the Latent Semantic Analysis (Deerwester et al., 1990), and

2. context co-occurrence probabilities were computed for all words, and the degree of combination of context co-occurrence probabilities and N-gram probabilities was computed for each word, depending on its distribution over the set of documents.

As for the first point, we utilized the computationally-light, iteration-based procedure. One reason for this is that the computational cost of SVD is very high when millions or more documents are processed. Furthermore, considering an extension of our model with a cognitive viewpoint, we believe an iteration-based model seems more reasonable than an algebraic model such as SVD.

As for the second point, we doubt the appropriateness to use the word's distribution as a measure of combination of two models. What we need to do is to distinguish words to which semantics should be considered and other words. We judged the distinction of content words and function words is good enough for that purpose, and developed their trigram-based distinction as shown in Figure 4.

Several topic-based models have been proposed based on the observation that certain words tend to have different probability distributions in different topics. For example, Florian and Yarowsky (1999) proposed the following model:

$$P(w_i|w_1^{i-1}) = \sum_t P(t|w_1^i) \cdot P_t(w_i|w_{i-N+1}^{i-1}) \quad (9)$$

where t denotes a topic id. Topics are obtained by hierarchical clustering from a training corpus, and a topic-specific language model, P_t, is learned from the clustered documents. Reductions in perplexity relative to a bigram model were 10.5% for the entire text and 33.5% for the target vocabulary.

Topic-based models capture long-distance lexical dependencies via intermediate topics. In other words, the estimated distribution of topics, $P(t|w_1^i)$, is the representation of a context. Our model does not use such intermediate topics, but accesses word co-occurrence information directly and represents a context as the accumulation of this information.

6 Conclusion

In this paper we described a novel language model of incorporating long-distance lexical dependencies based on context co-occurrence vectors. Reduced vector representation of word co-occurrences enables rather simple but effective representation of the context. Significant reductions in perplexity are obtained relative to a standard trigram model, both on the entire text (5.0%) and on the target vocabulary (27.2%).

Acknowledgments

The research described in this paper was supported in part by JSPS-RFTF96P00502 (The Japan Society for the Promotion of Science, Research for the Future Program).

References

Noah Coccaro and Daniel Jurafsky. 1998. Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of ICSLP-98*, volume 6, pages 2403–2406.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Radu Florian and David Yarowsky. 1999. Dynamic nonlocal language modeling via hierarchical topic-based adaptation. In *Proceedings of the 37rd Annual Meeting of ACL*, pages 167–174.

Pascale FUNG, LIU Xiaohu, and CHEUNG Chi Shun. 1999. Mixed language query disambiguation. In *Proceedings of the 37rd Annual Meeting of ACL*, pages 333–340.

Mard R. Ilgen and David A. Rushall. 1996. Recent advances in HNC's context vector information retrieval technology. In *TIPSTER PROGRAM PHASE II*, pages 149–158.

R. Kuhn and R. de Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.

R. Lau, Ronald Rosenfeld, and Salim Roukos. 1993. Trigger based language models: a maximum entropy approach. In *Proceedings of ICASSP*, pages 45–48.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution : Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of ACL*, pages 88–995.