Robust Approach to Abbreviating Terms: A Discriminative Latent Variable Model with Global Information

Xu Sun[†], **Naoaki Okazaki**[†], **Jun'ichi Tsujii**[†‡§]
[†]Department of Computer Science, University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan
[‡]School of Computer Science, University of Manchester, UK
[§]National Centre for Text Mining, UK
{sunxu, okazaki, tsujii}@is.s.u-tokyo.ac.jp

Abstract

The present paper describes a robust approach for abbreviating terms. First, in order to incorporate non-local information into abbreviation generation tasks, we present both implicit and explicit solutions: the latent variable model, or alternatively, the label encoding approach with global information. Although the two approaches compete with one another, we demonstrate that these approaches are also complementary. By combining these two approaches, experiments revealed that the proposed abbreviation generator achieved the best results for both the Chinese and English languages. Moreover, we directly apply our generator to perform a very different task from tradition, the abbreviation recognition. Experiments revealed that the proposed model worked robustly, and outperformed five out of six state-of-the-art abbreviation recognizers.

1 Introduction

Abbreviations represent fully expanded forms (e.g., *hidden markov model*) through the use of shortened forms (e.g., *HMM*). At the same time, abbreviations increase the ambiguity in a text. For example, in computational linguistics, the acronym *HMM* stands for *hidden markov model*, whereas, in the field of biochemistry, *HMM* is generally an abbreviation for *heavy meromyosin*. Associating abbreviations with their fully expanded forms is of great importance in various NLP applications (Pakhomov, 2002; Yu et al., 2006; HaCohen-Kerner et al., 2008).

The core technology for abbreviation disambiguation is to recognize the abbreviation definitions in the actual text. Chang and Schütze (2006) reported that 64,242 new abbreviations were introduced into the biomedical literatures in 2004. As such, it is important to maintain sense inventories (lists of abbreviation definitions) that are updated with the neologisms. In addition, based on the one-sense-per-discourse assumption, the recognition of abbreviation definitions assumes senses of abbreviations that are locally defined in a document. Therefore, a number of studies have attempted to model the generation processes of abbreviations: e.g., inferring the abbreviating mechanism of the *hidden markov model* into *HMM*.

An obvious approach is to manually design rules for abbreviations. Early studies attempted to determine the generic rules that humans use to intuitively abbreviate given words (Barrett and Grems, 1960; Bourne and Ford, 1961). Since the late 1990s, researchers have presented various methods by which to extract abbreviation definitions that appear in actual texts (Taghva and Gilbreth, 1999; Park and Byrd, 2001; Wren and Garner, 2002; Schwartz and Hearst, 2003; Adar, 2004; Ao and Takagi, 2005). For example, Schwartz and Hearst (2003) implemented a simple algorithm that mapped all alpha-numerical letters in an abbreviation to its expanded form, starting from the end of both the abbreviation and its expanded forms, and moving from right to left.

These studies performed highly, especially for English abbreviations. However, a more extensive investigation of abbreviations is needed in order to further improve definition extraction. In addition, we cannot simply transfer the knowledge of the hand-crafted rules from one language to another. For instance, in English, abbreviation characters are preferably chosen from the initial and/or capital characters in their full forms, whereas some

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 905–913,
Suntec, Singapore, 2-7 August 2009.

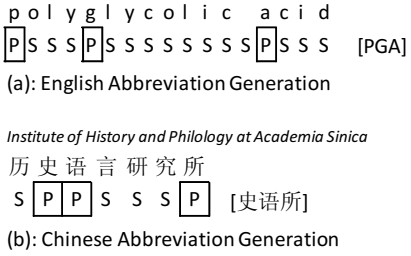

Figure 1: English (a) and Chinese (b) abbreviation generation as a sequential labeling problem.

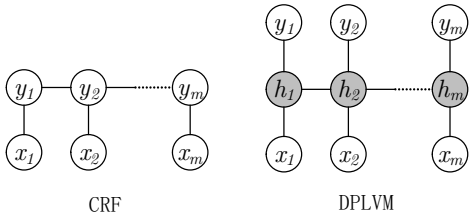

Figure 2: CRF *vs.* DPLVM. Variables x, y, and h represent observation, label, and latent variables, respectively.

other languages, including Chinese and Japanese, do not have word boundaries or case sensitivity.

A number of recent studies have investigated the use of machine learning techniques. Tsuruoka et al. (2005) formalized the processes of abbreviation generation as a sequence labeling problem. In the present study, each character in the expanded form is tagged with a label, $y \in \{\mathtt{P}, \mathtt{S}\}$[1], where the label *P produces the current character* and the label *S skips the current character*. In Figure 1 (a), the abbreviation *PGA* is generated from the full form *polyglycolic acid* because the underlined characters are tagged with P labels. In Figure 1 (b), the abbreviation is generated using the 2nd and 3rd characters, skipping the subsequent three characters, and then using the 7th character.

In order to formalize this task as a sequential labeling problem, we have assumed that the label of a character is determined by the local information of the character and its previous label. However, this assumption is not ideal for modeling abbreviations. For example, the model cannot make use of the number of words in a full form to determine and generate a suitable number of letters for the abbreviation. In addition, the model would be able to recognize the abbreviating process in Figure 1 (a) more reasonably if it were able to segment the word *polyglycolic* into smaller regions, e.g., *poly-glycolic*. Even though humans may use global or non-local information to abbreviate words, previous studies have not incorporated this information into a sequential labeling model.

In the present paper, we propose implicit and explicit solutions for incorporating non-local information. The implicit solution is based on the discriminative probabilistic latent variable model (DPLVM) in which non-local information is modeled by latent variables. We manually encode non-local information into the labels in order to provide an explicit solution. We evaluate the models on the task of *abbreviation generation*, in which a model produces an abbreviation for a given full form. Experimental results indicate that the proposed models significantly outperform previous abbreviation generation studies. In addition, we apply the proposed models to the task of *abbreviation recognition*, in which a model extracts the abbreviation definitions in a given text. To the extent of our knowledge, this is the first model that can perform both abbreviation generation and recognition at the state-of-the-art level, across different languages and with a simple feature set.

2 Abbreviator with Non-local Information

2.1 A Latent Variable Abbreviator

To implicitly incorporate non-local information, we propose discriminative probabilistic latent variable models (DPLVMs) (Morency et al., 2007; Petrov and Klein, 2008) for abbreviating terms. The DPLVM is a natural extension of the CRF model (see Figure 2), which is a special case of the DPLVM, with only one latent variable assigned for each label. The DPLVM uses latent variables to capture additional information that may not be expressed by the observable labels. For example, using the DPLVM, a possible feature could be "the current character $x_i = \mathtt{X}$, the label $y_i = \mathtt{P}$, and the latent variable $h_i = \mathtt{LV}$." The non-local information can be effectively modeled in the DPLVM, and the additional information at the previous position or many of the other positions in the past could be transferred via the latent variables (see Figure 2).

[1] Although the original paper of Tsuruoka et al. (2005) attached case sensitivity information to the P label, for simplicity, we herein omit this information.

Using the label set $Y = \{\texttt{P}, \texttt{S}\}$, abbreviation generation is formalized as the task of assigning a sequence of labels $\mathbf{y} = y_1, y_2, \ldots, y_m$ for a given sequence of characters $\mathbf{x} = x_1, x_2, \ldots, x_m$ in an expanded form. Each label, y_j, is a member of the possible labels Y. For each sequence, we also assume a sequence of latent variables $\mathbf{h} = h_1, h_2, \ldots, h_m$, which are unobservable in training examples.

We model the conditional probability of the label sequence $P(\mathbf{y}|\mathbf{x})$ using the DPLVM,

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h}, \mathbf{x}, \mathbf{\Theta}) P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}). \quad (1)$$

Here, $\mathbf{\Theta}$ represents the parameters of the model.

To ensure that the training and inference are efficient, the model is often restricted to have disjointed sets of latent variables associated with each label (Morency et al., 2007). Each h_j is a member in a set $\mathbf{H}_{y_j}$ of possible latent variables for the label y_j. Here, $\mathbf{H}$ is defined as the set of all possible latent variables, i.e., $\mathbf{H}$ is the union of all $\mathbf{H}_{y_j}$ sets. Since the sequences having $\mathbf{h}_j \notin \mathbf{H}_{y_j}$ will, by definition, yield $P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = 0$, the model is rewritten as follows (Morency et al., 2007; Petrov and Klein, 2008):

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h} \in \mathbf{H}_{y_1} \times \ldots \times \mathbf{H}_{y_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}). \quad (2)$$

Here, $P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta})$ is defined by the usual formulation of the conditional random field,

$$P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}) = \frac{\exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})}{\sum_{\forall \mathbf{h}} \exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})}, \quad (3)$$

where $\mathbf{f}(\mathbf{h}, \mathbf{x})$ represents a feature vector.

Given a training set consisting of n instances, $(\mathbf{x}_i, \mathbf{y}_i)$ (for $i = 1 \ldots n$), we estimate the parameters $\mathbf{\Theta}$ by maximizing the regularized log-likelihood,

$$L(\mathbf{\Theta}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{\Theta}) - R(\mathbf{\Theta}). \quad (4)$$

The first term expresses the conditional log-likelihood of the training data, and the second term represents a regularizer that reduces the overfitting problem in parameter estimation.

2.2 Label Encoding with Global Information

Alternatively, we can design the labels such that they explicitly incorporate non-local information.

Figure 3: Comparison of the proposed label encoding method with global information (GI) and the conventional label encoding method.

In this approach, the label y_i at position i attaches the information of the abbreviation length generated by its previous labels, $y_1, y_2, \ldots, y_{i-1}$. Figure 3 shows an example of a Chinese abbreviation. In this encoding, a label not only contains the *produce or skip* information, but also the abbreviation-length information, i.e., the label includes the number of all $\texttt{P}$ labels preceding the current position. We refer to this method as *label encoding with global information* (hereinafter *GI*). The concept of using label encoding to incorporate non-local information was originally proposed by Peshkin and Pfeffer (2003).

Note that the model-complexity is increased only by the increase in the number of labels. Since the length of the abbreviations is usually quite short (less than five for Chinese abbreviations and less than 10 for English abbreviations), the model is still tractable even when using the GI encoding.

The implicit (DPLVM) and explicit (GI) solutions address the same issue concerning the incorporation of non-local information, and there are advantages to combining these two solutions. Therefore, we will combine the implicit and explicit solutions by employing the GI encoding in the DPLVM (DPLVM+GI). The effects of this combination will be demonstrated through experiments.

2.3 Feature Design

Next, we design two types of features: language-independent features and language-specific features. Language-independent features can be used for abbreviating terms in English and Chinese. We use the features from #1 to #3 listed in Table 1.

Feature templates #4 to #7 in Table 1 are used for Chinese abbreviations. Templates #4 and #5 express the *Pinyin* reading of the characters, which represents a Romanization of the sound. Templates #6 and #7 are designed to detect character duplication, because identical characters will normally be skipped in the abbreviation process. On

#1	The input char. x_{i-1} and x_i
#2	Whether x_j is a numeral, for $j = (i-3)\dots i$
#3	The char. bigrams starting at $(i-2)\dots i$
#4	The *Pinyin* of char. x_{i-1} and x_i
#5	The *Pinyin* bigrams starting at $(i-2)\dots i$
#6	Whether $x_j = x_{j+1}$, for $j = (i-2)\dots i$
#7	Whether $x_j = x_{j+2}$, for $j = (i-3)\dots i$
#8	Whether x_j is uppercase, for $j = (i-3)\dots i$
#9	Whether x_j is lowercase, for $j = (i-3)\dots i$
#10	The char. 3-grams starting at $(i-3)\dots i$
#11	The char. 4-grams starting at $(i-4)\dots i$

Table 1: Language-independent features (#1 to #3), Chinese-specific features (#4 through #7), and English-specific features (#8 through #11).

the other hand, such duplication detection features are not so useful for English abbreviations.

Feature templates #8–#11 are designed for English abbreviations. Features #8 and #9 encode the orthographic information of expanded forms. Features #10 and #11 represent a contextual n-gram with a large window size. Since the number of letters in Chinese (more than $10K$ characters) is much larger than the number of letters in English (26 letters), in order to avoid a possible overfitting problem, we did not apply these feature templates to Chinese abbreviations.

Feature templates are instantiated with values that occur in positive training examples. We used all of the instantiated features because we found that the low-frequency features also improved the performance.

3 Experiments

For Chinese abbreviation generation, we used the corpus of Sun et al. (2008), which contains 2,914 abbreviation definitions for training, and 729 pairs for testing. This corpus consists primarily of noun phrases (38%), organization names (32%), and verb phrases (21%). For English abbreviation generation, we evaluated the corpus of Tsuruoka et al. (2005). This corpus contains 1,200 aligned pairs extracted from MEDLINE biomedical abstracts (published in 2001). For both tasks, we converted the aligned pairs of the corpora into labeled full forms and used the labeled full forms as the training/evaluation data.

The evaluation metrics used in the abbreviation generation are *exact-match accuracy* (hereinafter *accuracy*), including top-1 accuracy, top-2 accuracy, and top-3 accuracy. The top-N accuracy represents the percentage of correct abbreviations that are covered, if we take the top N candidates from the ranked labelings of an abbreviation generator.

We implemented the DPLVM in C++ and optimized the system to cope with large-scale problems. We employ the feature templates defined in Section 2.3, taking into account these 81,827 features for the Chinese abbreviation generation task, and the 50,149 features for the English abbreviation generation task.

For numerical optimization, we performed a gradient descent with the Limited-Memory BFGS (L-BFGS) optimization technique (Nocedal and Wright, 1999). L-BFGS is a second-order Quasi-Newton method that numerically estimates the curvature from previous gradients and updates. With no requirement on specialized Hessian approximation, L-BFGS can handle large-scale problems efficiently. Since the objective function of the DPLVM model is non-convex, different parameter initializations normally bring different optimization results. Therefore, to approach closer to the global optimal point, it is recommended to perform multiple experiments on DPLVMs with random initialization and then select a good start point. To reduce overfitting, we employed a L_2 Gaussian weight prior (Chen and Rosenfeld, 1999), with the objective function: $L(\mathbf{\Theta}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{\Theta}) - ||\mathbf{\Theta}||^2/\sigma^2$. During training and validation, we set $\sigma = 1$ for the DPLVM generators. We also set four latent variables for each label, in order to make a compromise between accuracy and efficiency.

Note that, for the label encoding with global information, many label transitions (e.g., $\mathtt{P}_2\mathtt{S}_3$) are actually impossible: the label transitions are strictly constrained, i.e., $y_i y_{i+1} \in \{\mathtt{P_jS_j}, \mathtt{P_jP_{j+1}}, \mathtt{S_jP_{j+1}}, \mathtt{S_jS_j}\}$. These constraints on the model topology (forward-backward lattice) are enforced by giving appropriate features a weight of $-\infty$, thereby forcing all forbidden labelings to have zero probability. Sha and Pereira (2003) originally proposed this concept of implementing transition restrictions.

4 Results and Discussion

4.1 Chinese Abbreviation Generation

First, we present the results of the Chinese abbreviation generation task, as listed in Table 2. To evaluate the impact of using latent variables, we chose the baseline system as the DPLVM, in which each label has only one latent variable. Since this

Model	T1A	T2A	T3A	Time
Heu (S08)	41.6	N/A	N/A	N/A
HMM (S08)	46.1	N/A	N/A	N/A
SVM (S08)	62.7	80.4	87.7	1.3 h
CRF	64.5	81.1	88.7	0.2 h
CRF+GI	66.8	82.5	90.0	0.5 h
DPLVM	67.6	83.8	91.3	0.4 h
DPLVM+GI (*)	**72.3**	**87.6**	**94.9**	1.1 h

Table 2: Results of Chinese abbreviation generation. T1A, T2A, and T3A represent top-1, top-2, and top-3 accuracy, respectively. The system marked with the * symbol is the recommended system.

special case of the DPLVM is exactly the CRF (see Section 2.1), this case is hereinafter denoted as the CRF. We compared the performance of the DPLVM with the CRFs and other baseline systems, including the heuristic system (Heu), the HMM model, and the SVM model described in *S08*, i.e., Sun et al. (2008). The heuristic method is a simple rule that produces the initial character of each word to generate the corresponding abbreviation. The SVM method described by Sun et al. (2008) is formalized as a regression problem, in which the abbreviation candidates are scored and ranked.

The results revealed that the latent variable model significantly improved the performance over the CRF model. All of its top-1, top-2, and top-3 accuracies were consistently better than those of the CRF model. Therefore, this demonstrated the effectiveness of using the latent variables in Chinese abbreviation generation.

As the case for the two alternative approaches for incorporating non-local information, the latent variable method and the label encoding method competed with one another (see DPLVM vs. CRF+GI). The results showed that the latent variable method outperformed the GI encoding method by +0.8% on the top-1 accuracy. The reason for this could be that the label encoding approach is a solution without the adaptivity on different instances. We will present a detailed discussion comparing DPLVM and CRF+GI for the English abbreviation generation task in the next subsection, where the difference is more significant.

In contrast, to a larger extent, the results demonstrate that these two alternative approaches are complementary. Using the GI encoding further improved the performance of the DPLVM (with +4.7% on top-1 accuracy). We found that major

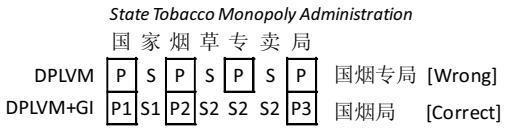

Figure 4: An example of the results.

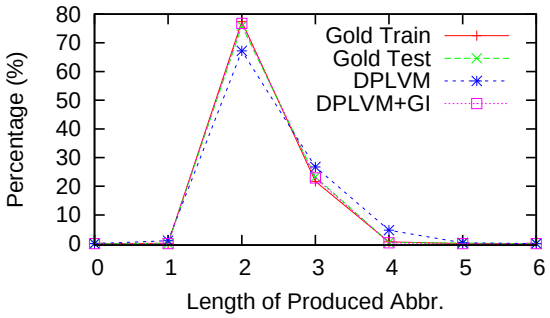

Figure 5: Percentage distribution of Chinese abbreviations/Viterbi-labelings grouped by length.

improvements were achieved through the more exact control of the output length. An example is shown in Figure 4. The DPLVM made correct decisions at three positions, but failed to control the abbreviation length.[2] The DPLVM+GI succeeded on this example. To perform a detailed analysis, we collected the statistics of the length distribution (see Figure 5) and determined that the GI encoding improved the abbreviation length distribution of the DPLVM.

In general, the results indicate that all of the sequential labeling models outperformed the SVM regression model with less *training time*.[3] In the SVM regression approach, a large number of negative examples are explicitly generated for the training, which slowed the process.

The proposed method, the latent variable model with GI encoding, is 9.6% better with respect to the top-1 accuracy compared to the best system on this corpus, namely, the SVM regression method. Furthermore, the top-3 accuracy of the latent variable model with GI encoding is as high as 94.9%, which is quite encouraging for practical usage.

4.2 English Abbreviation Generation

In the English abbreviation generation task, we randomly selected 1,481 instances from the gen-

[2]The Chinese abbreviation with $length = 4$ should have a very low probability, e.g., only 0.6% of abbreviations with $length = 4$ in this corpus.

[3]On Intel Dual-Core Xeon 5160/3 GHz CPU, excluding the time for feature generation and data input/output.

Model	T1A	T2A	T3A	Time
CRF	55.8	65.1	70.8	0.3 h
CRF+GI	52.7	63.2	68.7	1.3 h
CRF+GIB	56.8	66.1	71.7	1.3 h
DPLVM	57.6	67.4	73.4	0.6 h
DPLVM+GI	53.6	63.2	69.2	2.5 h
DPLVM+GIB (*)	**58.3**	N/A	N/A	3.0 h

Table 3: Results of English abbreviation generation.

```
    somatosensory evoked potentials
(a) P1P2          P3     P4       P5 SMEPS
(b) P             P      P        P  SEPS
(a): CRF+GI with p=0.001     [Wrong]
(b): DPLVM  with p=0.191     [Correct]
```

Figure 6: A result of "CRF+GI vs. DPLVM". For simplicity, the S labels are masked.

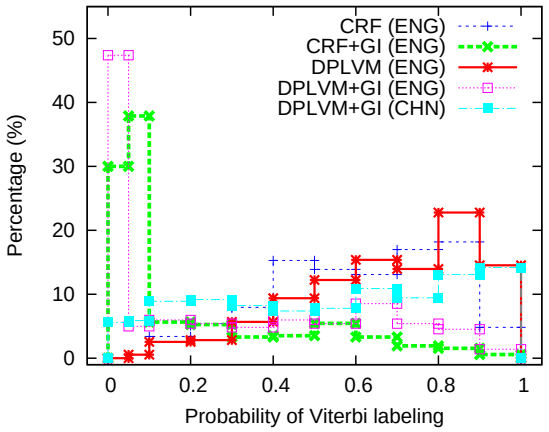

Figure 7: For various models, the probability distributions of the produced abbreviations on the test data of the English abbreviation generation task.

```
          mitomycin C
   DPLVM  P         P    MC   [Wrong]
DPLVM+GI  P1   P2    P3   MMC  [Correct]
```

Figure 8: Example of abbreviations composed of non-initials generated by the DPLVM and the DPLVM+GI.

eration corpus for training, and 370 instances for testing. Table 3 shows the experimental results. We compared the performance of the DPLVM with the performance of the CRFs. Whereas the use of the latent variables still significantly improves the generation performance, using the GI encoding undermined the performance in this task. In comparing the implicit and explicit solutions for incorporating non-local information, we can see that the implicit approach (the DPLVM) performs much better than the explicit approach (the GI encoding). An example is shown in Figure 6. The CRF+GI produced a Viterbi labeling with a low probability, which is an incorrect abbreviation. The DPLVM produced the correct labeling.

To perform a systematic analysis of the superior-performance of DPLVM compare to CRF+GI, we collected the probability distributions (see Figure 7) of the Viterbi labelings from these models ("DPLVM vs. CRF+GI" is highlighted). The curves suggest that the data sparseness problem could be the reason for the differences in performance. A large percentage (37.9%) of the Viterbi labelings from the CRF+GI (ENG) have very small probability values ($p < 0.1$). For the DPLVM (ENG), there were only a few (0.5%) Viterbi labelings with small probabilities. Since English abbreviations are often longer than Chinese abbreviations ($length < 10$ in English, whereas $length < 5$ in Chinese[4]), using the GI encoding resulted in a larger label set in English. Hence, the features become more sparse than in the Chinese case.[5] Therefore, a significant number of features could have been inadequately trained, resulting in Viterbi labelings with low probabilities. For the latent variable approach, its curve demonstrates that it did not cause a severe data sparseness problem.

The aforementioned analysis also explains the poor performance of the DPLVM+GI. However, the DPLVM+GI can actually produce correct abbreviations with 'believable' probabilities (high probabilities) in some 'difficult' instances. In Figure 8, the DPLVM produced an incorrect labeling for the difficult long form, whereas the DPLVM+GI produced the correct labeling containing non-initials.

Hence, we present a simple voting method to better combine the latent variable approach with the GI encoding method. We refer to this new combination as GI encoding with 'back-off' (hereinafter *GIB*): when the abbreviation generated by the DPLVM+GI has a 'believable' probability ($p > 0.3$ in the present case), the DPLVM+GI then outputs it. Otherwise, the system 'backs-off'

[4]See the curve DPLVM+GI (CHN) in Figure 7, which could explain the good results of GI encoding for the Chinese task.

[5]In addition, the training data of the English task is much smaller than for the Chinese task, which could make the models more sensitive to data sparseness.

Model	T1A	Time
CRF+GIB	67.2	0.6 h
DPLVM+GIB (*)	**72.5**	1.4 h

Table 4: Re-evaluating Chinese abbreviation generation with GIB.

Model	T1A
Heu (T05)	47.3
MEMM (T05)	55.2
DPLVM (*)	**57.5**

Table 5: Results of English abbreviation generation with five-fold cross validation.

to the parameters trained *without* the GI encoding (i.e., the DPLVM).

The results in Table 3 demonstrate that the DPVLM+GIB model significantly outperformed the other models because the DPLVM+GI model improved the performance in some 'difficult' instances. The DPVLM+GIB model was robust even when the data sparseness problem was severe.

By re-evaluating the DPLVM+GIB model for the previous Chinese abbreviation generation task, we demonstrate that the back-off method also improved the performance of the Chinese abbreviation generators ($+0.2\%$ from DPLVM+GI; see Table 4).

Furthermore, for interests, like Tsuruoka et al. (2005), we performed a five-fold cross-validation on the corpus. Concerning the training time in the cross validation, we simply chose the DPLVM for comparison. Table 5 shows the results of the DPLVM, the heuristic system (Heu), and the maximum entropy Markov model (MEMM) described by Tsuruoka et al. (2005).

5 Recognition as a Generation Task

We directly migrate this model to the abbreviation recognition task. We simplify the abbreviation recognition to a restricted generation problem (see Figure 9). When a context expression (CE) with a parenthetical expression (PE) is met, the recognizer generates the Viterbi labeling for the CE, which leads to the PE or NULL. Then, if the Viterbi labeling leads to the PE, we can, at the same time, use the labeling to decide the full form within the CE. Otherwise, NULL indicates that the PE is *not* an abbreviation.

For example, in Figure 9, the recognition is restricted to a generation task with five possible la-

```
... cannulate for arterial pressure (AP)...
(1)                        P  P          AP
(2)                  P        P          AP
(3)  P                        P          AP
(4)        P                  P          AP
(5) SSSSSSSSSSSSSSSSSSSSSSSSSSSSS  NULL
```

Figure 9: Abbreviation recognition as a restricted generation problem. In some labelings, the S labels are masked for simplicity.

Model	P	R	F
Schwartz & Hearst (SH)	**97.8**	94.0	95.9
SaRAD	89.1	91.9	90.5
ALICE	96.1	92.0	94.0
Chang & Schütze (CS)	94.2	90.0	92.1
Nadeau & Turney (NT)	95.4	87.1	91.0
Okazaki et al. (OZ)	97.3	**96.9**	**97.1**
CRF	89.8	94.8	92.1
CRF+GI	93.9	97.8	95.9
DPLVM	92.5	97.7	95.1
DPLVM+GI (*)	**94.2**	**98.1**	**96.1**

Table 6: Results of English abbreviation recognition.

belings. Other labelings are impossible, because they will generate an abbreviation that is not *AP*. If the first or second labeling is generated, *AP* is selected as an abbreviation of *arterial pressure*. If the third or fourth labeling is generated, then *AP* is selected as an abbreviation of *cannulate for arterial pressure*. Finally, the fifth labeling (NULL) indicates that *AP* is *not* an abbreviation.

To evaluate the recognizer, we use the corpus[6] of Okazaki et al. (2008), which contains 864 abbreviation definitions collected from 1,000 MEDLINE scientific abstracts. In implementing the recognizer, we simply use the model from the abbreviation generator, with the same feature templates (31,868 features) and training method; the major difference is in the restriction (according to the PE) of the decoding stage and penalizing the probability values of the NULL labelings[7].

For the evaluation metrics, following Okazaki et al. (2008), we use precision ($P = k/m$), recall ($R = k/n$), and the F-score defined by

[6]The previous abbreviation generation corpus is improper for evaluating recognizers, and there is no related research on this corpus. In addition, there has been no report of Chinese abbreviation recognition because there is no data available. The previous generation corpus (Sun et al., 2008) is improper because it lacks local contexts.

[7]Due to the data imbalance of the training corpus, we found the probability values of the NULL labelings are abnormally high. To deal with this imbalance problem, we simply penalize all NULL labelings by using $p = p - 0.7$.

Model	P	R	F
CRF+GIB	94.0	98.9	96.4
DPLVM+GIB	**94.5**	**99.1**	**96.7**

Table 7: English abbreviation recognition with back-off.

$2PR/(P+R)$, where k represents #instances in which the system extracts correct full forms, m represents #instances in which the system extracts the full forms regardless of correctness, and n represents #instances that have annotated full forms. Following Okazaki et al. (2008), we perform 10-fold cross validation.

We prepared six state-of-the-art abbreviation recognizers as baselines: Schwartz and Hearst's method (SH) (2003), SaRAD (Adar, 2004), ALICE (Ao and Takagi, 2005), Chang and Schütze's method (CS) (Chang and Schütze, 2006), Nadeau and Turney's method (NT) (Nadeau and Turney, 2005), and Okazaki et al.'s method (OZ) (Okazaki et al., 2008). Some methods use implementations on the web, including SH[8], CS[9], and ALICE[10]. The results of other methods, such as SaRAD, NT, and OZ, are reproduced for this corpus based on their papers (Okazaki et al., 2008).

As can be seen in Table 6, using the latent variables significantly improved the performance (see DPLVM vs. CRF), and using the GI encoding improved the performance of both the DPLVM and the CRF. With the F-score of 96.1%, the DPLVM+GI model outperformed five of six state-of-the-art abbreviation recognizers. Note that all of the six systems were specifically designed and optimized for this recognition task, whereas the proposed model is directly transported from the generation task. Compared with the generation task, we find that the F-measure of the abbreviation recognition task is much higher. The major reason for this is that there are far fewer classification candidates of the abbreviation recognition problem, as compared to the generation problem.

For interests, we also tested the effect of the GIB approach. Table 7 shows that the back-off method further improved the performance of both the DPLVM and the CRF model.

[8]http://biotext.berkeley.edu/software.html

[9]http://abbreviation.stanford.edu/

[10]http://uvdb3.hgc.jp/ALICE/ALICE_index.html

6 Conclusions and Future Research

We have presented the DPLVM and GI encoding by which to incorporate non-local information in abbreviating terms. They were competing and generally the performance of the DPLVM was superior. On the other hand, we showed that the two approaches were complementary. By combining these approaches, we were able to achieve state-of-the-art performance in abbreviation generation and recognition in the same model, across different languages, and with a simple feature set. As discussed earlier herein, the training data is relatively small. Since there are numerous unlabeled full forms on the web, it is possible to use a semi-supervised approach in order to make use of such raw data. This is an area for future research.

Acknowledgments

We thank Yoshimasa Tsuruoka for providing the English abbreviation generation corpus. We also thank the anonymous reviewers who gave helpful comments. This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan).

References

Eytan Adar. 2004. SaRAD: A simple and robust abbreviation dictionary. *Bioinformatics*, 20(4):527–533.

Hiroko Ao and Toshihisa Takagi. 2005. ALICE: An algorithm to extract abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 12(5):576–586.

June A. Barrett and Mandalay Grems. 1960. Abbreviating words systematically. *Communications of the ACM*, 3(5):323–324.

Charles P. Bourne and Donald F. Ford. 1961. A study of methods for systematically abbreviating english words and names. *Journal of the ACM*, 8(4):538–552.

Jeffrey T. Chang and Hinrich Schütze. 2006. Abbreviations in biomedical text. In Sophia Ananiadou and John McNaught, editors, *Text Mining for Biology and Biomedicine*, pages 99–119. Artech House, Inc.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. *Technical Report CMU-CS-99-108, CMU*.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2008. Combined one sense disambiguation of abbreviations. In *Proceedings of ACL'08: HLT, Short Papers*, pages 61–64, June.

Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. 2007. Latent-dynamic discriminative models for continuous gesture recognition. *Proceedings of CVPR'07*, pages 1–8.

David Nadeau and Peter D. Turney. 2005. A supervised learning approach to acronym identification. In *the 8th Canadian Conference on Artificial Intelligence (AI'2005) (LNAI 3501)*, page 10 pages.

Jorge Nocedal and Stephen J. Wright. 1999. Numerical optimization. *Springer*.

Naoaki Okazaki, Sophia Ananiadou, and Jun'ichi Tsujii. 2008. A discriminative alignment model for abbreviation recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 657–664, Manchester, UK.

Serguei Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of ACL'02*, pages 160–167.

Youngja Park and Roy J. Byrd. 2001. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of EMNLP'01*, pages 126–133.

Leonid Peshkin and Avi Pfeffer. 2003. Bayesian information extraction network. In *Proceedings of IJCAI'03*, pages 421–426.

Slav Petrov and Dan Klein. 2008. Discriminative log-linear grammars with latent variables. *Proceedings of NIPS'08*.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *the 8th Pacific Symposium on Biocomputing (PSB'03)*, pages 451–462.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. *Proceedings of HLT/NAACL'03*.

Xu Sun, Houfeng Wang, and Bo Wang. 2008. Predicting chinese abbreviations from definitions: An empirical learning approach using support vector regression. *Journal of Computer Science and Technology*, 23(4):602–611.

Kazem Taghva and Jeff Gilbreth. 1999. Recognizing acronyms and their definitions. *International Journal on Document Analysis and Recognition (IJDAR)*, 1(4):191–198.

Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. A machine learning approach to acronym generation. In *Proceedings of the ACL-ISMB Workshop*, pages 25–31.

Jonathan D. Wren and Harold R. Garner. 2002. Heuristics for identification of acronym-definition patterns within text: towards an automated construction of comprehensive acronym-definition dictionaries. *Methods of Information in Medicine*, 41(5):426–434.

Hong Yu, Won Kim, Vasileios Hatzivassiloglou, and John Wilbur. 2006. A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations. *ACM Transactions on Information Systems (TOIS)*, 24(3):380–404.