Predicting Chinese Abbreviations with Minimum Semantic Unit and Global Constraints

Longkai Zhang Li Li Houfeng Wang Xu Sun
Key Laboratory of Computational Linguistics (Peking University)
Ministry of Education, China
zhlongk@qq.com, {li.l,wanghf,xusun}@pku.edu.cn

Abstract

We propose a new Chinese abbreviation prediction method which can incorporate rich local information while generating the abbreviation globally. Different to previous character tagging methods, we introduce the minimum semantic unit, which is more fine-grained than character but more coarse-grained than word, to capture word level information in the sequence labeling framework. To solve the "character duplication" problem in Chinese abbreviation prediction, we also use a substring tagging strategy to generate local substring tagging candidates. We use an integer linear programming (ILP) formulation with various constraints to globally decode the final abbreviation from the generated candidates. Experiments show that our method outperforms the state-of-the-art systems, without using any extra resource.

1 Introduction

Abbreviation is defined as a shortened description of the original fully expanded form. For example, "NLP" is the abbreviation for the corresponding full form "Natural Language Processing". The existence of abbreviations makes it difficult to identify the terms conveying the same concept in the information retrieval (IR) systems and machine translation (MT) systems. Therefore, it is important to maintain a dictionary of the prevalent original full forms and the corresponding abbreviations.

Previous works on Chinese abbreviation generation focus on the sequence labeling method, which give each character in the full form an extra label to indicate whether it is kept in the abbreviation. One drawback of the character tagging strategy is that Chinese characters only contain limited amount of information. Using character-based method alone is not enough for Chinese abbreviation generation. Intuitively we can think of a word as the basic tagging unit to incorporate more information. However, if the basic tagging unit is word, we need to design lots of tags to represent which characters are kept for each unit. For a word with n characters, we should design at least 2^n labels to cover all possible situations. This reduces the generalization ability of the proposed model. Besides, the Chinese word segmentation errors may also hurt the performance. Therefore we propose the idea of "Minimum Semantic Unit" (MSU) which is the minimum semantic unit in Chinese language. Some of the MSUs are words, while others are more fine-grained than words. The task of selecting representative characters in the full form can be further broken down into selecting representative characters in the MSUs. We model this using the MSU-based tagging method, which can both utilize semantic information while keeping the tag set small.

Meanwhile, the sequence labeling method performs badly when the "character duplication" phenomenon exists. Many Chinese long phrases contain duplicated characters, which we refer to as the "character duplication" phenomenon. There is no sound criterion for the character tagging models to decide which of the duplicated character should be kept in the abbreviation and which one to be skipped. An example is "北京航空航天大学"(Beijing University of Aeronautics and Astronautics) whose abbreviation is "北航". The character "航" appears twice in the full form and only one is kept in the abbreviation. In these cases, we can break the long phase into local substrings. We can find the representative characters in the substrings instead of the long full form and let the decoding phase to integrate useful information globally. We utilize this sub-string based approach and obtain this local tagging information by labeling

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1405–1414,
October 25-29, 2014, Doha, Qatar.

on the sub-string of the full character sequence.

Given the MSU-based and substring-based methods mentioned above, we can get a list of potential abbreviation candidates. Some of these candidates may not agree on keeping or skipping of some specific characters. To integrate their advantages while considering the consistency, we further propose a global decoding strategy using Integer Linear Programming(ILP). The constraints in ILP can naturally incorporate 'non-local' information in contrast to probabilistic constraints that are estimated from training examples. We can also use linguistic constraints like "adjacent identical characters is not allowed" to decode the correct abbreviation in examples like the previous "北航" example.

Experiments show that our Chinese abbreviation prediction system outperforms the state-of-the-art systems. In order to reduce the size of the search space, we further propose pruning constraints that are learnt from the training corpus. Experiment shows that the average number of constraints is reduced by about 30%, while the top-1 accuracy is not affected.

The paper is structured as follows. Section 1 gives the introduction. In section 2 we describe our method, including the MSUs, the substring-based tagging strategy and the ILP decoding process. Experiments are described in section 3. We also give a detailed analysis of the results in section 3. In section 4 related works are introduced, and the paper is concluded in the last section.

2 System Architecture

2.1 Chinese Abbreviation Prediction

Chinese abbreviations are generated by selecting representative characters from the full forms. For example, the abbreviation of "北京大学" (Peking University) is "北大" which is generated by selecting the first and third characters, see TABLE 1. This can be tackled from the sequence labeling point of view.

Full form	北	京	大	学
Status	Keep	Skip	Keep	Skip
Result	北		大	

Table 1: The abbreviation "北大" of the full form "北京大学" (Peking University)

From TABLE 1 we can see that Chinese abbreviation prediction is a problem of selecting representative characters from the original full form[1]. Based on this assumption, previous works mainly focus on this character tagging schema. In these methods, the basic tagging unit is the Chinese character. Each character in the full form is labeled as 'K' or 'S', where 'K' means the current character should be kept in abbreviation and 'S' means the current character should be skipped.

However, a Chinese character can only contain limited amount of information. Using character-based method alone is not enough for Chinese abbreviation generation. We introduce an MSU-based method, which models the process of selecting representative characters given local MSU information.

2.2 MSU Based Tagging

2.2.1 Minimum Semantic Unit

Because using the character-based method is not enough for Chinese abbreviation generation, we may think of word as the basic tagging unit to incorporate more information intuitively. In English, the abbreviations (similar to acronyms) are usually formed by concatenating initial letters or parts of a series of words. In other words, English abbreviation generation is based on words in the full form. However, in Chinese, word is not the most suitable abbreviating unit. Firstly, there is no natural boundary between Chinese words. Errors from the Chinese word segmentation tools will accumulate to harm the performance of abbreviation prediction. Second, it is hard to design a reasonable tag set when the length of a possible Chinese word is very long. The second column of TABLE 2 shows different ways of selecting representative characters of Chinese words with length 3. For a Chinese compound word with 3 characters, there are 6 possible ways to select characters. In this case we should have at least 6 kinds of tags to cover all possible situations. The case is even worse for words with more complicated structures. A suitable abbreviating unit should be smaller than word.

We propose the "Minimum Semantic Unit (MSU)" as the basic tagging unit. We define MSU as follows:

1. A word whose length is less or equal to 2 is an MSU.

[1] A small portion of Chinese abbreviations are not generated from the full form. For example, the abbreviation of "山东"(Shan Dong Province) is "鲁". However, we can use a look-up table to get this kind of abbreviations.

Full form	SK Label	MSUs
幼儿园(nursery)	幼/K 儿/S 园/S	幼儿+园
补贴费(allowance)	补/S 贴/K 费/S	补贴+费
信用卡(Credit card)	信/S 用/S 卡/K	信用+卡
水电站(Hydropower Station)	水/K 电/K 站/S	水+电+站
参议院(Senate)	参/K 议/S 院/K	参+议+院
音乐团(Music group)	音/S 乐/K 团/K	音乐+团

Table 2: Representing characters of Chinese words with length 3 (K for keep and S for skip) and the corresponding MSUs

2. A word whose length is larger than 2, but does not contain any MSUs with length equal to 2. For example, "火车站"(Railway Station) is not an MSU because the first two characters "火车"(Train) can form an MSU.

By this definition, all 6 strings in TABLE 2 are often thought as a word, but they are not MSUs in our view. Their corresponding MSU forms are shown in TABLE 2.

We collect all the MSUs from the benchmark datasets provided by the second International Chinese Word Segmentation Bakeoff[2]. We choose the Peking University (PKU) data because it is more fine-grained than all other corpora. Suppose we represent the segmented data as L (In our case L is the PKU word segmentation data), the MSU selecting algorithm is shown in TABLE 3.

For a given full form, we first segment it using a standard word segmenter to get a coarse-grained segmentation result. Here we use the Stanford Chinese Word Segmenter [3]. Then we use the MSU set to segment each word using the strategy of "Maximum Forward Matching"[4] to get the fine-grained MSU segmentation result.

2.2.2 Labeling strategy

For MSU-based tagging, we use a labeling method which uses four tags, "KSFL". "K" stands for "Keep the whole unit", "S" stands for "Skip the whole unit", "F" stands for "keep the First character of the unit", and Label "L" stands for "keep the Last character of the unit". An example is shown in TABLE 4.

The "KSFL" tag set is also applicable for MSUs whose length is greater than 2 (an example is "巧克力/chocolate"). By examining the corpus we find that such MSUs are either kept of skipped in the final abbreviations. Therefore, the labels of these long MSUs are either 'K' or 'S'. Empirically, this assumption holds for MSUs, but does not hold for words[5].

"国家语言文字工作委员会" (The abbreviation is "国家语委")	
KSFL	国家/K 语言/F 文字/S 工作/S 委员/F 会/S

Table 4: The abbreviation "国家语委" of "国家语言文字工作会" (National Linguistics Work Committee) based on MSU tagging.

2.2.3 Feature templates

The feature templates we use are as follows. See TABLE 5.

1. Word X_i $(-2 \leq i \leq 2)$
2. POS tag of word X_i $(-2 \leq i \leq 2)$
3. Word Bigrams (X_i, X_{i+1}) $(-2 \leq i \leq 1)$
4. Type of word X_i $(-2 \leq i \leq 2)$
5. Length of word X_i $(-2 \leq i \leq 2)$

Table 5: Feature templates for unit tagging. X represents the MSU sequence of the full form. X_i represents the ith MSU in the sequence.

Templates 1, 2 and 3 express word uni-grams and bi-grams. In MSU-based tagging, we can utilize the POS information, which we get from the Stanford Chinese POS Tagger[6]. In template 4, the type of word refers to whether it is a number, an English word or a Chinese word. Because the basic tagging unit is MSU, which carries word information, we can use many features that are infeasible in character-based tagging.

[2] http://www.sighan.org/bakeoff2005/

[3] http://nlp.stanford.edu/software/segmenter.shtml

[4] In Chinese, "Forward" means from left to right.

[5] In table 2, all examples are partly kept.

[6] http://nlp.stanford.edu/software/tagger.shtml

Init:
Let $MSUSet$ = empty set
For each word w in L:
  If $Length(w) \leq 2$
    Add w to $MSUSet$
  End if
End for
For each word w in L:
  If $Length(w) > 2$ and no word x in $MSUSet$ is a substring of w
    Add w to $MSUSet$
  End if
End for
Return $MSUSet$

Table 3: Algorithm for collecting MSUs from the PKU corpus

2.2.4 Sequence Labeling Model

The MSU-based method gives each MSU an extra indicative label. Therefore any sequence labeling model is appropriate for the method. Previous works showed that Conditional Random Fields (CRFs) can outperform other sequence labeling models like MEMMs in abbreviation generation tasks (Sun et al., 2009; Tsuruoka et al., 2005). For this reason we choose CRFs model in our system.

For a given full form's MSU list, many candidate abbreviations are generated by choosing the k-best results of the CRFs. We can use the forward-backward algorithm to calculate the probability of a specified tagging result. To reduce the searching complexity in the ILP decoding process, we delete those candidate tagged sequences with low probability.

2.3 Substring Based Tagging

As mentioned in the introduction, the sequence labeling method, no matter character-based or MSU-based, perform badly when the "character duplication" phenomenon exists. When the full form contains duplicated characters, there is no sound criterion for the sequence tagging strategy to decide which of the duplicated character should be kept in the abbreviation and which one to be skipped. On the other hand, we can tag the substrings of the full form to find the local representative characters in the substrings of the long full form. Therefore, we propose the sub-string based approach to given labeling results on sub-strings. These results can be integrated into a more accurate result using ILP constraints, which we will describe in the next section.

Another reason for using the sub-string based methods is that long full forms contain more characters and are much easier to make mistakes during the sequence labeling phase. Zhang et al. (2012) shows that if the full form contains less than 5 characters, a simple tagger can reach an accuracy of 70%. Zhang et al. (2012) also shows that if the full form is longer than 10 characters, the average accuracy is less than 30%. The numerous potential candidates make it hard for the tagger to choose the correct one. For the long full forms, although the whole sequence is not correctly labeled, we find that if we only consider its short substrings, we may find the correct representative characters. This information can be integrated into the decoding model to adjust the final result.

We use the MSU-based tagging method in the sub-string tagging. The labeling strategy and feature templates are the same to the MSU-based tagging method. In practice, enumerating all subsequences of a given full form is infeasible if the full form is very long. For a given full form, we use the boundary MSUs to reduce the possible sub-sequence set. For example, "中国科学院"(Chinese Academy of Science) has 5 subsequences: "中国", "中国科学", "科学", "科学院" and "院".

2.4 ILP Formulation of Decoding

Given the MSU-based and sub-sequence-based methods mentioned above as well as the prevalent character-based methods, we can get a list of potential abbreviation candidates and abbreviated substrings. We should integrate their advantages while keeping the consistency between each

candidate. Therefore we further propose a global decoding strategy using Integer Linear Programming(ILP). The constraints in ILP can naturally incorporate 'non-local' information in contrast to probabilistic constraints that are estimated from training examples. We can also use linguistic constraints like "adjacent identical characters is not allowed" to decode the correct abbreviation in examples like the "北航" example in section 1.

Formally, given the character sequence of the full form $c = c_1...c_l$, we keep Q top-ranked MSU-based tagging results T=($T_1, ..., T_Q$) and M tagged substrings S=($S_1, ..., S_M$) using the methods described in previous sections. We also use N top-ranked character-based tagging results R=($R_1, ..., R_N$) based on the previous character-based works. We also define the set $U = S \cup R \cup T$ as the union of all candidate sequences. Our goal is to find an optimal binary variable vector solution $\vec{v} = \vec{x}\vec{y}\vec{z} = (x_1, ..., x_M, y_1, ..., y_N, z_1, ..., z_Q)$ that maximizes the object function:

$$\lambda_1 \sum_{i=1}^{M} score(S_i) \cdot x_i + \lambda_2 \sum_{i=1}^{N} score(R_i) \cdot y_i$$

$$+\lambda_3 \sum_{i=1}^{Q} score(T_i) \cdot z_i$$

subject to constrains in TABLE 6. The parameters λ_1, λ_2, λ_3 controls the preference of the three parts, and can be decided using cross-validation.

Constraint 1 indicates that x_i, y_i, z_i are all boolean variables. They are used as indicator variables to show whether the corresponding tagged sequence is in accordance with the final result.

Constraint 2 is used to guarantee that at most one candidate from the character-based tagging is preserved. We relax the constraint to allow the sum to be zero in case that none of the top-ranked candidate is suitable to be the final result. If the sum equals zero, then the sub-sequence based tagging method will generate a more suitable result. Constrain 3 has the same utility for the MSU-based tagging.

Constraint 4, 5, 6 are inter-method constraints. We use them to guarantee that the labels of the preserved sequences of different tagging methods do not conflict with each other. Constraint 7 is used to guarantee that the labels of the preserved sub-strings do not conflict with each other.

Constraint 8 is used to solve the "character duplicate" problem. When two identical characters are kept adjacently, only one of them will be kept. Which one will be kept depends on the global decoding score. This is the advantage of ILP against traditional sequence labeling methods.

2.5 Pruning Constraints

The efficiency of solving the ILP decoding problem depends on the number of candidate tagging sequences N and Q, as well as the number of subsequences M. Usually, N and Q is less than 10 in our experiment. Therefore, M influences the time complexity the most. Because we use the boundary of MSUs instead of enumerating all possible subsequences, the value of M can be largely reduced.

Some characters are always labeled as "S" or "K" once the context is given. We can use this phenomenon to reduce the search space of decoding. Let c_i denote the i_{th} character relative to the current character c_0 and t_i denote the tag of c_i. The context templates we use are listed in TABLE 7.

Uni-gram Contexts	c_0, c_{-1}, c_1
Bi-gram Contexts	$c_{-1}c0$, $c_{-1}c_1$, c_0c_1

Table 7: Context templates used in pruning

With respect to a training corpus, if a context C relative to c_0 always assigns a certain tag t to c_0, then we can use this constraint in pruning. We judge the degree of "always" by checking whether $\frac{count(C \wedge t_0 = t)}{count(C)} > threshold$. The threshold is a non-negative real number under 1.0.

3 Experiments

3.1 Data and Evaluation Metric

We use the abbreviation corpus provided by Institute of Computational Linguistics (ICL) of Peking University in our experiments. The corpus is similar to the corpus used in Sun et al. (2008, 2009); Zhang et al. (2012). It contains $8,015$ Chinese abbreviations, including noun phrases, organization names and some other types. Some examples are presented in TABLE 8. We use 80% abbreviations as training data and the rest as testing data. In some cases, a long phrase may contain more than one abbreviation. For these cases, the corpus just keeps their most commonly used abbreviation for each full form.

The evaluation metric used in our experiment is the top-K accuracy, which is also used by Tsuruoka et al. (2005), Sun et al. (2009) and

1. $x_i \in \{0,1\}$, $y_i \in \{0,1\}$, $z_i \in \{0,1\}$
2. $\sum_{i=1}^{N} y_i \leq 1$
3. $\sum_{i=1}^{Q} z_i \leq 1$
4. $\forall R_i \in R$, $S_j \in S$, if R_i and S_j have a same position but the position gets different labels, then $y_i + x_j \leq 1$
5. $\forall T_i \in T$, $S_j \in S$, if T_i and S_j have a same position but the position gets different labels, then $z_i + x_j \leq 1$
6. $\forall R_i \in R$, $T_j \in T$, if R_i and T_j have a same position but the position gets different labels, then $x_i + z_j \leq 1$
7. $\forall S_i, S_j \in S$ if S_i and S_j have a same position but the position gets different labels, then $z_i + z_j \leq 1$
8. $\forall S_i, S_j \in S$ if the last character S_i keeps is the same as the first character S_j keeps, then $z_i + z_j \leq 1$

Table 6: Constraints for ILP

Type	Full form	Abbreviation
Noun Phrase	优秀稿件(Excellent articles)	优稿
Organization	作家协会(Writers' Association)	作协
Coordinate phrase	受伤死亡(Injuries and deaths)	伤亡
Proper noun	传播媒介(Media)	传媒

Table 8: Examples of the corpus (Noun Phrase, Organization, Coordinate Phrase, Proper Noun)

Zhang et al. (2012). The top-K accuracy measures what percentage of the reference abbreviations are found if we take the top N candidate abbreviations from all the results. In our experiment, top-10 candidates are considered in re-ranking phrase and the measurement used is top-1 accuracy (which is the accuracy we usually refer to) because the final aim of the algorithm is to detect the exact abbreviation.

CRF++[7] , an open source linear chain CRF tool, is used in the sequence labeling part. For ILP part, we use lpsolve[8], which is also an open source tool. The parameters of these tools are tuned through cross-validation on the training data.

3.2 Results

TABLE 9 shows the top-K accuracy of the character-based and MSU-based method. We can see that the MSU-based tagging method can utilize word information, which can get better performance than the character-based method. We can also figure out that the top-5 candidates include the reference abbreviation for most full forms. Therefore reasonable decoding by considering all possible labeling of sequences may improve the performance. Although the MSU-based methods only outperforms character-based methods by 0.75% for top-1 accuracy, it is much better when considering top-2 to top-5 accuracy (+2.5%). We further select the top-ranked candidates for ILP decoding. Therefore the MSU-based method can further improve the performance in the global decoding phase.

K	char-based	MSU-based
1	0.5714	0.5789
2	0.6879	0.7155
3	0.7681	0.7819
4	0.8070	0.8283
5	0.8333	**0.8583**

Table 9: Top-K ($K \leq 5$) results of character-based tagging and MSU-based tagging

We then use the top-5 candidates of character-based method and MSU-based method, as well as the top-2 results of sub-sequence labeling in the ILP decoding phase. Then we select the top-ranked candidate as the final abbreviation of each instance. TABLE 10 shows the results. We can see that the accuracy of our method is 61.0%, which improved by +3.89% compared to the character-based method, and +3.14% compared to the MSU-based method.

We find that the ILP decoding phase do play an important role in generating the right an-

[7] http://crfpp.sourceforge.net/

[8] http://lpsolve.sourceforge.net/5.5/

Method	Top-1 Accuracy
Char-based	0.5714
MSU-based	0.5789
ILP Result	**0.6103**

Table 10: Top-1 Accuracy after ILP decoding

swer. Some reference abbreviations which are not picked out by either tagging method can be found out after decoding. TABLE 11 shows the example of the organization name "高等学校统一招生办公室" (Higher Education Admissions Office). Neither the character-based method nor the MSU-based method finds the correct answer "高招办", while after ILP decoding, "高招办" becomes the final result. TABLE 12 and TABLE 13 give two more examples.

True Result	高招办
Char-based	高办
MSU-based	高统办
ILP Decoding	高招办

Table 11: Top-1 result of "高等学校统一招生办公室" (Higher Education Admissions Office)

True Result	超值
Char-based	物值
MSU-based	物超值
ILP Decoding	超值

Table 12: Top-1 result of "物超价值" (Articles exceed the value)

True Result	声光视效
Char-based	声光效
MSU-based	声效果
ILP Decoding	声光视效

Table 13: Top-1 result of "声音灯光视觉效果" (Visual effects of sound and lights)

3.3 Improvements Considering Length

Full forms that are longer than five characters are long terms. Long terms contain more characters, which is much easier to make mistakes. Figure 1 shows the top-1 accuracy respect to the term length using different tagging methods and using ILP decoding. The x-axis represents the length of the full form. The y-axis represents top-1 accuracy. We find that our method works especially better than pure character-based or MSU-based approach when the full form is long. By decoding using ILP, both local and global information are incorporated. Therefore many of these errors can be eliminated.

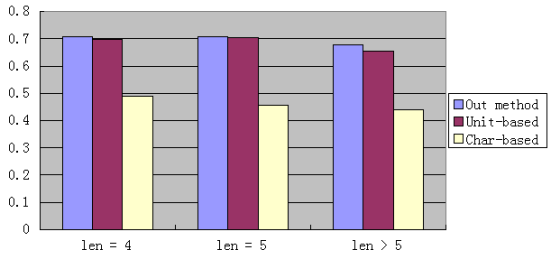

Figure 1: Top-1 accuracy of different methods considering length

3.4 Effect of pruning

As discussed in previous sections, if we are able to pre-determine that some characters in a certain context should be kept or skipped, then the number of possible boolean variable x can be reduced. TABLE 14 shows the differences. To guarantee a high accuracy, we set the threshold to be 0.99. When the original full form is partially tagged by the pruning constraints, the number of boolean variables per full form is reduced from 34.4 to 25.5. By doing this, we can improve the prediction speed over taking the raw input.

From TABLE 14 we can also see that the top-1 accuracy is not affected by these pruning constraints. This is obvious, because CRF itself has a strong modeling ability. The pruning constraints cannot improve the model accuracy. But they can help eliminate those false candidates to make the ILP decoding faster.

	Accuracy	Average length	Time(s)
raw	0.6103	34.4	12.5
pruned	0.6103	25.5	7.1

Table 14: Comparison of testing time of raw input and pruned input

3.5 Compare with the State-of-the-art Systems

We also compare our method with previous methods, including Sun et al. (2009) and Zhang et al. (2012). Because we use a different corpus, we re-implement the system Sun et al. (2009), Zhang

et al. (2012) and Sun et al. (2013), and experiment on our corpus. The first two are CRF+GI and DPLVM+GI in Sun et al. (2009), which are reported to outperform the methods in Tsuruoka et al. (2005) and Sun et al. (2008). For DPLVM we use the same model in Sun et al. (2009) and experiment on our own data. We also compare our approach with the method in Zhang et al. (2012). However, Zhang et al. (2012) uses different sources of search engine result information to re-rank the original candidates. We do not use any extra web resources. Because Zhang et al. (2012) uses web information only in its second stage, we use "BIEP"(the tag set used by Zhang et al. (2012)) to denote the first stage of Zhang et al. (2012), which also uses no web information. TABLE 15 shows the results of the comparisons. We can see that our method outperforms all other methods which use no extra resource. Because Zhang et al. (2012) uses extra web resource, the top-1 accuracy of Zhang et al. (2012) is slightly better than ours.

Method	Top-1 Accuracy
CRF+GI	0.5850
DPLVM+GI	0.5990
BIEP	0.5812
Zhang et al. (2012)	**0.6205**
Our Result	**0.6103**

Table 15: Comparison with the state-of-the-art systems

4 Related Work

Previous research mainly focuses on "abbreviation disambiguation", and machine learning approaches are commonly used (Park and Byrd, 2001; HaCohen-Kerner et al., 2008; Yu et al., 2006; Ao and Takagi, 2005). These ways of linking abbreviation pairs are effective, however, they cannot solve our problem directly. In many cases the full form is definite while we don't know the corresponding abbreviation.

To solve this problem, some approaches maintain a database of abbreviations and their corresponding "full form" pairs. The major problem of pure database-building approach is obvious. It is impossible to cover all abbreviations, and the building process is quit laborious. To find these pairs automatically, a powerful approach is to find the reference for a full form given the context, which is referred to as "abbreviation generation".

There is research on heuristic rules for generating abbreviations Barrett and Grems (1960); Bourne and Ford (1961); Taghva and Gilbreth (1999); Park and Byrd (2001); Wren et al. (2002); Hearst (2003). Most of them achieved high performance. However, hand-crafted rules are time consuming to create, and it is not easy to transfer the knowledge of rules from one language to another.

Recent studies of abbreviation generation have focused on the use of machine learning techniques. Sun et al. (2008) proposed a supervised learning approach by using SVM model. Tsuruoka et al. (2005); Sun et al. (2009) formalized the process of abbreviation generation as a sequence labeling problem. In Tsuruoka et al. (2005) each character in the full form is associated with a binary value label y, which takes the value S (Skip) if the character is not in the abbreviation, and value P (Preserve) if the character is in the abbreviation. Then a MEMM model is used to model the generating process. Sun et al. (2009) followed this schema but used DPLVM model to incorporate both local and global information, which yields better results. Sun et al. (2013) also uses machine learning based methods, but focuses on the negative full form problem, which is a little different from our work.

Besides these pure statistical approaches, there are also many approaches using Web as a corpus in machine learning approaches for generating abbreviations.Adar (2004) proposed methods to detect such pairs from biomedical documents. Jain et al. (2007) used web search results as well as search logs to find and rank abbreviates full pairs, which show good result. The disadvantage is that search log data is only available in a search engine backend. The ordinary approaches do not have access to search engine internals. Zhang et al. (2012) used web search engine information to rerank the candidate abbreviations generated by statistical approaches. Compared to their approaches, our method uses no extra resource, but reaches comparable results.

ILP shows good results in many NLP tasks. Punyakanok et al. (2004); Roth and Yih (2005) used it in semantic role labeling (SRL). Martins et al. (2009) used it in dependency parsing. (Zhao and Marcus, 2012) used it in Chinese word segmentation. (Riedel and Clarke, 2006) used ILP

in dependency parsing. However, previous works mainly focus on the constraints of avoiding boundary confliction. For example, in SRL, two argument of cannot overlap. In CWS, two Chinese words cannot share a same character. Different to their methods, we investigate on the conflict of labels of character sub-sequences.

5 Conclusion and Future work

We propose a new Chinese abbreviation prediction method which can incorporate rich local information while generating the abbreviation globally. We propose the MSU, which is more coarse-grained than character but more fine-grained than word, to capture word information in the sequence labeling framework. Besides the MSU-based method, we use a substring tagging strategy to generate local substring tagging candidates. We use an ILP formulation with various constraints to globally decode the final abbreviation from the generated candidates. Experiments show that our method outperforms the state-of-the-art systems, without using any extra resource. This method is not limited to Chinese abbreviation generation, it can also be applied to similar languages like Japanese.

The results are promising and outperform the baseline methods. The accuracy can still be improved. Potential future works may include using semi-supervised methods to incorporate unlabeled data and design reasonable features from large corpora. We are going to study on these issues in the future.

Acknowledgments

This research was partly supported by National Natural Science Foundation of China (No.61370117,61333018,61300063),Major National Social Science Fund of China(No.12&ZD227), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), and Doctoral Fund of Ministry of Education of China (No. 20130001120004). The contact author of this paper, according to the meaning given to this role by Key Laboratory of Computational Linguistics, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, is Houfeng Wang. We thank Ke Wu for part of our work is inspired by his previous work at KLCL.

References

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

Ao, H. and Takagi, T. (2005). Alice: an algorithm to extract abbreviations from medline. *Journal of the American Medical Informatics Association*, 12(5):576–586.

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

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

HaCohen-Kerner, Y., Kass, A., and Peretz, A. (2008). Combined one sense disambiguation of abbreviations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 61–64. Association for Computational Linguistics.

Hearst, M. S. (2003). A simple algorithm for identifying abbreviation definitions in biomedical text.

Jain, A., Cucerzan, S., and Azzam, S. (2007). Acronym-expansion recognition and ranking on the web. In *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on*, pages 209–214. IEEE.

Martins, A. F., Smith, N. A., and Xing, E. P. (2009). Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 342–350. Association for Computational Linguistics.

Park, Y. and Byrd, R. (2001). Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 conference on empirical methods in natural language processing*, pages 126–133.

Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Compu-*

tational Linguistics, page 1346. Association for Computational Linguistics.

Riedel, S. and Clarke, J. (2006). Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 129–137. Association for Computational Linguistics.

Roth, D. and Yih, W.-t. (2005). Integer linear programming inference for conditional random fields. In *Proceedings of the 22nd international conference on Machine learning*, pages 736–743. ACM.

Sun, X., Li, W., Meng, F., and Wang, H. (2013). Generalized abbreviation prediction with negative full forms and its application on improving chinese web search. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 641–647, Nagoya, Japan. Asian Federation of Natural Language Processing.

Sun, X., Okazaki, N., and Tsujii, J. (2009). Robust approach to abbreviating terms: A discriminative latent variable model with global information. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 905–913. Association for Computational Linguistics.

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

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

Tsuruoka, Y., Ananiadou, S., and Tsujii, J. (2005). A machine learning approach to acronym generation. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 25–31. Association for Computational Linguistics.

Wren, J., Garner, H., et al. (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.

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

Zhang, L., Li, S., Wang, H., Sun, N., and Meng, X. (2012). Constructing Chinese abbreviation dictionary: A stacked approach. In *Proceedings of COLING 2012*, pages 3055–3070, Mumbai, India. The COLING 2012 Organizing Committee.

Zhao, Q. and Marcus, M. (2012). Exploring deterministic constraints: from a constrained english pos tagger to an efficient ilp solution to chinese word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1062, Jeju Island, Korea. Association for Computational Linguistics.