Semi-Supervised Chinese Word Segmentation Using Partial-Label Learning With Conditional Random Fields

Fan Yang
Nuance Communications Inc.
fan.yang@nuance.com

Paul Vozila
Nuance Communications Inc.
paul.vozila@nuance.com

Abstract

There is rich knowledge encoded in online web data. For example, punctuation and entity tags in Wikipedia data define some word boundaries in a sentence. In this paper we adopt partial-label learning with conditional random fields to make use of this valuable knowledge for semi-supervised Chinese word segmentation. The basic idea of partial-label learning is to optimize a cost function that marginalizes the probability mass in the constrained space that encodes this knowledge. By integrating some domain adaptation techniques, such as EasyAdapt, our result reaches an F-measure of 95.98% on the CTB-6 corpus, a significant improvement from both the supervised baseline and a previous proposed approach, namely *constrained decode*.

1 Introduction

A general approach for supervised Chinese word segmentation is to formulate it as a character sequence labeling problem, to label each character with its location in a word. For example, Xue (2003) proposes a four-label scheme based on some linguistic intuitions: 'B' for the beginning character of a word, 'I' for the internal characters, 'E' for the ending character, and 'S' for single-character word. Thus the word sequence "洽谈会很 成功" can be turned into a character sequence with labels as 洽\B 谈\I 会\E 很\S 成\B 功\E. A machine learning algorithm for sequence labeling, such as conditional random fields (CRF) (Lafferty et al., 2001), can be applied to the labelled training data to learn a model.

Labelled data for supervised learning of Chinese word segmentation, however, is usually expensive and tends to be of a limited amount. Researchers are thus interested in semi-supervised learning, which is to make use of unlabelled data to further improve the performance of supervised learning. There is a large amount of unlabelled data available, for example, the Gigaword corpus in the LDC catalog or the Chinese Wikipedia on the web.

Faced with the large amount of unlabelled data, an intuitive idea is to use self-training or EM, by first training a baseline model (from the supervised data) and then iteratively decoding the unlabelled data and updating the baseline model. Jiao et al. (2006) and Mann and McCallum (2007) further propose to minimize the entropy of the predicted label distribution on unlabeled data and use it as a regularization term in CRF (i.e. *entropy regularization*). Beyond these ideas, Liang (2005) and Sun and Xu (2011) experiment with deriving a large set of statistical features such as mutual information and accessor variety from unlabelled data, and add them to supervised discriminative training. Zeng et al. (2013b) experiment with graph propagation to extract information from unlabelled data to regularize the CRF training. Yang and Vozila (2013), Zhang et al. (2013), and Zeng et al. (2013a) experiment with co-training for semi-supervised Chinese word segmentation. All these approaches only leverage the distribution of the unlabelled data, yet do not make use of the knowledge that the unlabelled data might have integrated in.

There could be valuable information encoded within the unlabelled data that researchers can take advantage of. For example, punctuation creates natural word boundaries (Li and Sun, 2009): the character before a comma can only be labelled as either 'S' or 'E', while the character after a comma can only be labelled as 'S' or 'B'. Furthermore, entity tags (HTML tags or Wikipedia tags) on the web, such as emphasis and cross reference, also provide rich information for word segmentation: they might define a word or at least

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 90–98,
October 25-29, 2014, Doha, Qatar.

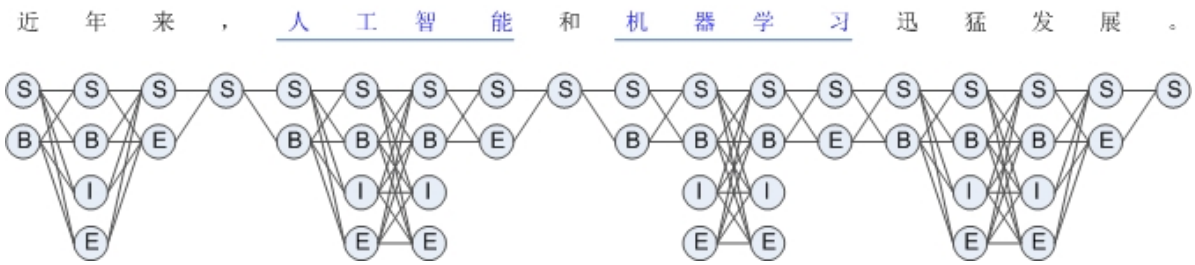

Figure 1: Sausage constraint (partial labels) from natural annotations and punctuation

give word boundary information similar to punctuation. Jiang et al. (2013) refer to such structural information on the web as natural annotations, and propose that they encode knowledge for NLP. For Chinese word segmentation, natural annotations and punctuation create a sausage[1] constraint for the possible labels, as illustrated in Figure 1. In the sentence "近年来，人工智能和机器学习迅猛发展。", the first character 近 can only be labelled with 'S' or 'B'; and the characters 来 before the comma and 展 before the Chinese period can only be labelled as 'S' or 'E'. "人工智能" and "机器学习" are two Wikipedia entities, and so they define the word boundaries before the first character and after the last character of the entities as well. The single character 和 between these two entities has only one label 'S'. This sausage constraint thus encodes rich information for word segmentation.

To make use of the knowledge encoded in the sausage constraint, Jiang et al. (2013) adopt a constrained decode approach. They first train a baseline model with labelled data, and then run constrained decode on the unlabelled data by binding the search space with the sausage; and so the decoded labels are consistent with the sausage constraint. The unlabelled data, together with the labels from constrained decode, are then selectively added to the labelled data for training the final model. This approach, using constrained decode as a middle step, provides an indirect way of leaning the knowledge. However, the middle step, constrained decode, has the risk of reinforcing the errors in the baseline model: the decoded labels added to the training data for building the final model might contain errors introduced from the baseline model. The knowledge encoded in the data carrying the information from punctuation and natural annotations is thus polluted by the errorful re-decoded labels.

A sentence where each character has exactly one label is fully-labelled; and a sentence where each character receives all possible labels is zero-labelled. A sentence with sausage-constrained labels can be viewed as partially-labelled. These partial labels carry valuable information that researchers would like to learn in a model, yet the normal CRF training typically uses fully-labelled sentences. Recently, Täckström et al. (2013) propose an approach to train a CRF model directly from partial labels. The basic idea is to marginalize the probability mass of the constrained sausage in the cost function. The normal CRF training using fully-labelled sentences is a special case where the sausage constraint is a linear line; while on the other hand a zero-labelled sentence, where the sausage constraint is the full lattice, makes no contribution in the learning since the sum of probabilities is deemed to be one. This new approach, without the need of using constrained re-decoding as a middle step, provides a direct means to learn the knowledge in the partial labels.

In this research we explore using the partial-label learning for semi-supervised Chinese word segmentation. We use the CTB-6 corpus as the labelled training, development and test data, and use the Chinese Wikipedia as the unlabelled data. We first train a baseline model with labelled data only, and then selectively add Wikipedia data with partial labels to build a second model. Because the Wikipedia data is out of domain and has distribution bias, we also experiment with two domain adaptation techniques: model interpolation and EasyAdapt (Daumé III, 2007). Our result reaches an F-measure of 95.98%, an absolute improvement of 0.72% over the very strong base-

[1] Also referred to as confusion network.

line (corresponding to 15.19% relative error reduction), and 0.33% over the constrained decode approach (corresponding to 7.59% relative error reduction). We conduct a detailed error analysis, illustrating how partial-label learning excels constrained decode in learning the knowledge encoded in the Wikipedia data. As a note, our result also out-performs (Wang et al., 2011) and (Sun and Xu, 2011).

2 Partial-Label Learning with CRF

In this section, we review in more detail the partial-label learning algorithm with CRF proposed by (Täckström et al., 2013). CRF is an exponential model that expresses the conditional probability of the labels given a sequence, as Equation 1, where y denotes the labels, x denotes the sequence, $\Phi(x, y)$ denotes the feature functions, and θ is the parameter vector. $Z(x) = \sum_y \exp(\theta^T \Phi(x, y))$ is the normalization term.

$$p_\theta(y|x) = \frac{\exp(\theta^T \Phi(x, y))}{Z(x)} \quad (1)$$

In full-label training, where each item in the sequence is labelled with exactly one tag, maximum likelihood is typically used as the optimization target. We simply sum up the log-likelihood of the n labelled sequences in the training set, as shown in Equation 2.

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{n} log\ p_\theta(y|x) \\ &= \sum_{i=1}^{n} (\theta^T \Phi(x_i, y_i) - log\ Z(x_i)) \end{aligned} \quad (2)$$

The gradient is calculated as Equation 3, in which the first term $\frac{1}{n}\sum_{i=1}^{n}\Phi_j$ is the empirical expectation of feature function Φ_j, and the second term $E[\Phi_j]$ is the model expectation. Typically a forward-backward process is adopted for calculating the latter.

$$\frac{\partial}{\partial\theta_j} L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\Phi_j - E[\Phi_j] \quad (3)$$

In partial-label training, each item in the sequence receives multiple labels, and so for each sequence we have a sausage constraint, denoted as $\hat{Y}(x, \tilde{y})$. The marginal probability of the sausage is defined as Equation 4.

$$p_\theta(\hat{Y}(x, \tilde{y})|x) = \sum_{y \in \hat{Y}(x,\tilde{y})} p_\theta(y|x) \quad (4)$$

The optimization target thus is to maximize the probability mass of the sausage, as shown in Equation 5.

$$L(\theta) = \sum_{i=1}^{n} log p_\theta(\hat{Y}(x_i, \tilde{y_i})|x_i) \quad (5)$$

A gradient-based approach such as L-BFGS (Liu and Nocedal, 1989) can be employed to optimize Equation 5. The gradient is calculated as Equation 6, where $E_{\hat{Y}(x,\tilde{y})}[\Phi_j]$ is the empirical expectation of feature function Φ_j constrained by the sausage, and $E[\Phi_j]$ is the same model expectation as in standard CRF. $E_{\hat{Y}(x,\tilde{y})}[\Phi_j]$ can be calculated via a forward-backward process in the constrained sausage.

$$\frac{\partial}{\partial\theta_j} L(\theta) = E_{\hat{Y}(x,\tilde{y})}[\Phi_j] - E[\Phi_j] \quad (6)$$

For fully-labelled sentences, $E_{\hat{Y}(x,\tilde{y})}[\Phi_j] = \frac{1}{n}\sum_{i=1}^{n}\Phi_j$, and so the standard CRF is actually a special case of the partial-label learning.

3 Experiment setup

In this section we describe the basic setup for our experiments of semi-supervised Chinese word segmentation.

3.1 Data

We use the CTB-6 corpus as the labelled data. We follow the official CTB-6 guideline in splitting the corpus into a training set, a development set, and a test set. The training set has 23420 sentences; the development set has 2079 sentences; and the test set has 2796 sentences. These are fully-labelled data.

For unlabelled data we use the Chinese Wikipedia. The Wikipedia data is quite noisy and asks for a lot of cleaning. We first filter out references and lists etc., and sentences with obviously bad segmentations, for example, where every character is separated by a space. We also remove sentences that contain mostly English words. We then convert all characters into full-width. We also convert traditional Chinese characters into simplified characters using the tool

mediawiki-zhconverter[2]. We then randomly select 7737 sentences and reserve them as the test set.

To create the partial labels in the Wikipedia data, we use the information from cross-reference, emphasis, and punctuation. In our pilot study we found that it's beneficial to force a cross-reference or emphasis entity as a word when the item has 2 or 3 characters. That is, if an entity in the Wikipedia has three characters it receives the labels of "BIE"; and if it has two characters it is labelled as "BE".[3]

3.2 Supervised baseline model

We create the baseline supervised model by using an order-1 linear CRF with L2 regularization, to label a character sequence with the four candidate labels "BIES". We use the tool wapiti (Lavergne et al., 2010).

Following Sun et al. (2009), Sun (2010), and Low et al. (2005), we extract two types of features: character-level features and word-level features. Given a character c_0 in the character sequence ...$c_{-2}c_{-1}c_0c_1c_2$...:

Character-level features :

- Character unigrams: c_{-2}, c_{-1}, c_0, c_1, c_2
- Character bigrams: $c_{-2}c_{-1}$, $c_{-1}c_{-0}$, c_0c_1, c_1c_2
- Consecutive character equivalence: $?c_{-2} = c_{-1}$, $?c_{-1} = c_{-0}$, $?c_0 = c_1$, $?c_1 = c_2$
- Separated character equivalence: $?c_{-3} = c_{-1}$, $?c_{-2} = c_0$, $?c_{-1} = c_1$, $?c_0 = c_2$, $?c_1 = c_3$
- Whether the current character is a punctuation: $?Punct(c_0)$
- Character sequence pattern: $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$. We classify all characters into four types. Type one has three characters '年' (year) '月' (month) '日' (date). Type two includes number characters. Type three includes English characters. All others are Type four characters. Thus "去年三月S" would generate the character sequence pattern "41213".

Word-level features :

- The identity of the string $c[s:i]$ ($i-6 < s < i$), if it matches a word from the list of word unigrams; multiple features could be generated.
- The identity of the string $c[i:e]$ ($i < e < i+6$), if it matches a word; multiple features could be generated.
- The identity of the bi-gram $c[s:i-1]c[i:e]$ ($i-6 < s, e < i+6$), if it matches a word bigram; multiple features could be generated.
- The identity of the bi-gram $c[s:i]c[i+1:e]$ ($i-6 < s, e < i+6$), if it matches a word bigram; multiple features could be generated.
- Idiom. We use the idiom list from (Sun and Xu, 2011). If the current character c_0 and its surrounding context compose an idiom, we generate a feature for c_0 of its position in the idiom. For example, if $c_{-1}c_0c_1c_2$ is an idiom, we generate feature "Idiom-2" for c_0.

The above features together with label bigrams are fed to wapiti for training. The supervised baseline model is created with the CTB-6 corpus without the use of Wikipedia data.

3.3 Partial-label learning

The overall process of applying partial-label learning to Wikipedia data is shown in Algorithm 1. Following (Jiang et al., 2013), we first train the supervised baseline model, and use it to estimate the potential contribution for each sentence in the Wikipedia training data. We label the sentence with the baseline model, and then compare the labels with the constrained sausage. For each character, a consistent label is defined as an element in the constrained labels. For example, if the constrained labels for a character are "SB", the label 'S' or 'B' is consistent but 'I' or 'E' is not. The number of inconsistent labels for each sentence is then used as its potential contribution to the partial-label learning: higher number indicates that the partial-labels for the sentence contain more knowledge that the baseline system does not integrate, and so have higher potential contribution. The Wikipedia training sentences are then ranked by their potential contribution, and the top

[2]https://github.com/tszming/mediawiki-zhconverter

[3]Another possibility is to label it as "SS" but we find that it's very rare the case.

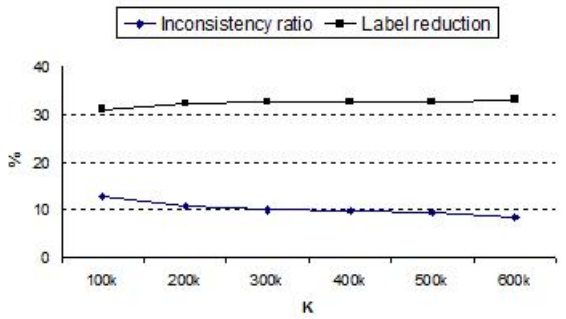

Figure 2: Encoded knowledge: inconsistency ratio and label reduction

K sentences together with their partial labels are then added to the CTB-6 training data to build a new model, using partial-label learning.[4] In our experiments, we try six data points with K = 100k, 200k, 300k, 400k, 500k, 600k. Figure 2 gives a rough idea of the knowledge encoded in Wikipedia for these data points with inconsistency ratio and label reduction. Inconsistency ratio is the percentage of characters that have inconsistent labels; and label reduction is the percentage of the labels reduced in the full lattice.

We modify wapiti to implement the partial-label learning as described in Section 2. Same as baseline, L2 regularization is adopted.

Algorithm 1 Partial-label learning

1. Train supervised baseline model M_0
2. For each sentence x in Wiki-Train:
3. $y \leftarrow$ Decode(x, M_0)
4. diff $\leftarrow$ Inconsistent(y, $\hat{Y}(x, \tilde{y})$)
5. if diff > 0:
6. C $\leftarrow$ C $\cup$ ($\hat{Y}(x, \tilde{y})$, diff)
7. Sort(C, diff, reverse)
8. Train model M^{pl} with CTB-6 and top K sentences in C using partial-label learning

3.4 Constrained decode

Jiang et al. (2013) implement the constrained decode algorithm with perceptron. However, CRF is generally believed to out-perform perceptron, yet the comparison of CRF vs perceptron is out of the scope of this paper. Thus for fair comparison, we re-implement the constrained decode algorithm with CRF.

Algorithm 2 shows the constrained decode implementation. We first train the baseline model with the CTB-6 data. We then use this baseline model to run normal decode and constrained decode for each sentence in the Wikipedia training set. If the normal decode and constrained decode have different labels, we add the constrained decode together with the number of different labels to the filtered Wikipedia training corpus. The filtered Wikipedia training corpus is then sorted using the number of different labels, and the top K sentences with constrained decoded labels are then added to the CTB-6 training data for building a new model using normal CRF.

Algorithm 2 Constrained decode

1. Train supervised baseline model M_0
2. For each sentence x in Wiki-Train:
3. $y \leftarrow$ Decode(x, M_0)
4. $\bar{y} \leftarrow$ ConstrainedDecode(x, M_0)
5. diff $\leftarrow$ Difference(y, $\bar{y}$)
6. if diff > 0:
7. C $\leftarrow$ C $\cup$ ($\bar{y}$, diff)
8. Sort(C, diff, reverse)
9. Train model M^{cd} with CTB-6 and top K sentences in C using normal CRF

4 Evaluation on Wikipedia test set

In order to determine how well the models learn the encoded knowledge (i.e. partial labels) from the Wikipedia data, we first evaluate the models against the Wikipedia test set. The Wikipedia test set, however, is only partially-labelled. Thus the metric we use here is *consistent label accuracy*, similar to how we rank the sentences in Section 3.3, defined as whether a predicted label for a character is an element in the constrained labels. Because partial labels are only sparsely distributed in the test data, a lot of characters receive all four labels in the constrained sausage. Evaluating against characters with all four labels do not really represent the models' difference as it is deemed to be consistent. Thus beyond evaluating against all characters in the Wikipedia test set (referred to as *Full* measurement), we also evaluate against characters that are only constrained with less than four labels (referred to as *Label* measurement). The *Label* measurement focuses on en-

[4] Knowledge is sparsely distributed in the Wikipedia data. Using the Wikipedia data without the CTB-6 data for partial-label learning does not necessarily guarantee convergence. Also the CTB-6 training data helps to learn that certain label transitions, such as "B B" or "E E", are not legal.

coded knowledge in the test set and so can better represent the model's capability of learning from the partial labels.

Results are shown in Figure 3 with the *Full* measurement and in Figure 4 with the *Label* measurement. The x axes are the size of Wikipedia training data, as explained in Section 3.3. As can be seen, both constrained decode and partial-label learning perform much better than the baseline supervised model that is trained from CTB-6 data only, indicating that both of them are learning the encoded knowledge from the Wikipedia training data. Also we see the trend that the performance improves with more data in training, also suggesting the learning of encoded knowledge. Most importantly, we see that partial-label learning consistently out-performs constrained decode in all data points. With the *Label* measurement, partial-label learning gives 1.7% or higher absolute improvement over constrained decode across all data points. At the data point of 600k, constrained decode gives an accuracy of 97.14%, while partial-label learning gives 98.93% (baseline model gives 87.08%). The relative gain (from learning the knowledge) of partial-label learning over constrained decode is thus 18% $((98.93 - 97.14)/(97.14 - 87.08))$. These results suggest that partial-label learning is more effective in learning the encoded knowledge in the Wikipedia data than constrained decode.

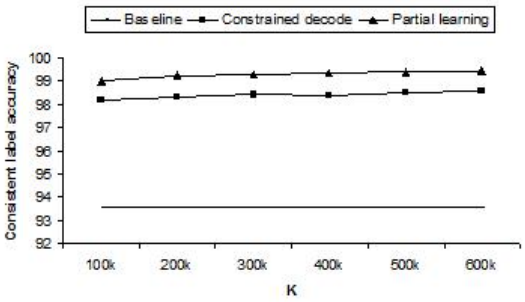

Figure 3: Wiki label evaluation results: Full

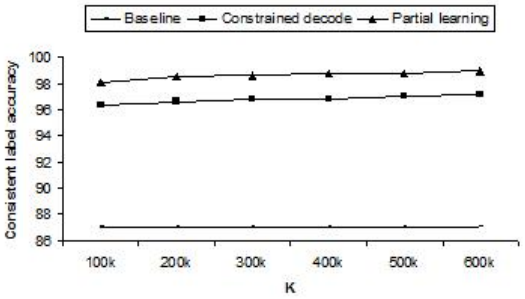

Figure 4: Wiki label evaluation results: Label

5 CTB evaluation

5.1 Model adaptation

Our ultimate goal, however, is to determine whether we can leverage the encoded knowledge in the Wikipedia data to improve the word segmentation in CTB-6. We run our models against the CTB-6 test set, with results shown in Figure 5. Because we have fully-labelled sentences in the CTB-6 data, we adopt the F-measure as our evaluation metric here. The baseline model achieves 95.26% in F-measure, providing a state-of-the-art supervised performance. Constrained decode is able to improve on this already very strong baseline performance, and we see the nice trend of higher performance with more unlabeled data for training, indicating that constrained decode is making use of the encoded knowledge in the Wikipedia data to help CTB-6 segmentation.

When we look at the partial-label model, however, the results tell a totally different story.

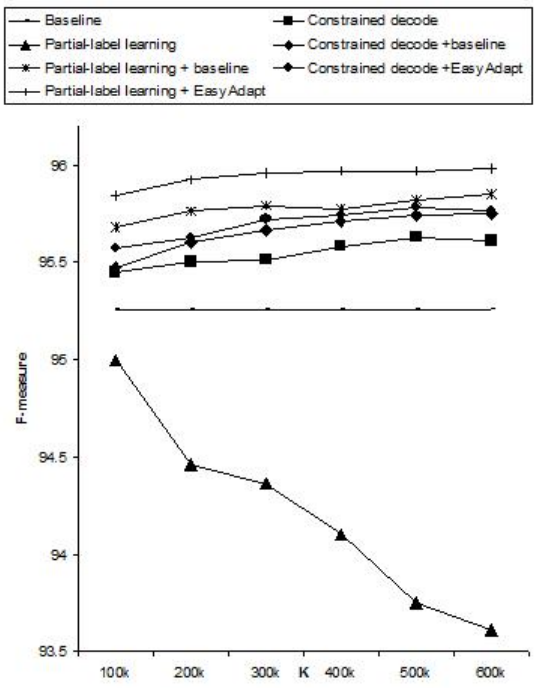

Figure 5: CTB evaluation results

First, it actually performs worse than the baseline model, and the more data added to training, the worse the performance is. In the previous section we show that partial-label learning is more effective in learning the encoded knowledge in Wikipedia data than constrained decode. So, what goes wrong? We hypothesize that there is an out-of-domain distribution bias in the partial labels, and so the more data we add, the worse the in-domain performance is. Constrained decode actually helps to smooth out the out-of-domain distribution bias by using the re-decoded labels with the in-domain supervised baseline model. For example, both the baseline model and constrained decode correctly give the segmentation "提供/了/运输/和/**给排水**/之/便", while partial-label learning gives incorrect segmentation "提供/了/运输/和/**给/排水**/之/便". Looking at the Wikipedia training data, 排水 is tagged as an entity 13 times; and 给排水, although occurs 13 times in the data, is never tagged as an entity. Partial-label learning, which focuses on the tagged entities, thus overrules the segmentation of 给排水. Constrained decode, on the other hand, by using the correctly re-decoded labels from the baseline model, observes enough evidence to correctly segment 给排水 as a word.

To smooth out the out-of-domain distribution bias, we experiment with two approaches: model interpolation and EasyAdapt (Daumé III, 2007).

5.1.1 Model interpolation

We linearly interpolate the model of partial-label learning M^{pl} with the baseline model M_0 to create the final model M_+^{pl}, as shown in Equation 7. The interpolation weight is optimized via a grid search between 0.0 and 1.0 with a step of 0.1, tuned on the CTB-6 development set. Again we modify wapiti so that it takes two models and an interpolation weight as input. For each model it creates a search lattice with posteriors, and then linearly combines the two lattices using the interpolation weight to create the final search space for decoding. As shown in Figure 5, model M_+^{pl} consistently out-performs constrained decode in all data points. We also see the trend of better performance with more training data.

$$M_+^{pl} = \lambda * M_0 + (1 - \lambda) * M^{pl} \qquad (7)$$

5.1.2 EasyAdapt

EasyAdapt is a straightforward technique but has been shown effective in many domain adaptation tasks (Daumé III, 2007). We train the model M_{ea}^{pl} with feature augmentation. For each out-of-domain training instance $< x_o, y_o >$, where x_o is the input features and y_o is the (partial) labels, we copy the features and file them as an additional feature set, and so the training instance becomes $< x_o, x_o, y_o >$. The in-domain training data remains the same. Consistent with (Daumé III, 2007), EasyAdapt gives us the best performance, as show in Figure 5. Furthermore, unlike in (Jiang et al., 2013) where they find a plateau, our results show no harm adding more training data for partial-label learning when integrated with domain adaptation, although the performance seems to saturate after 400k sentences.

Finally, we search for the parameter setting of best performance on the CTB-6 development set, which is to use EasyAdapt with $K = 600k$ sentences of Wikipedia data. With this setting, the performance on the CTB-6 test set is 95.98% in F-measure. This is 0.72% absolute improvement over supervised baseline (corresponding to 15.19% relative error reduction), and 0.33% absolute improvement over constrained decode (corresponding to 7.59% relative error reduction); the differences are both statistically significant ($p < 0.001$).[5] As a note, this result out-performs (Sun and Xu, 2011) (95.44%) and (Wang et al., 2011) (95.79%), and the differences are also statistically significant ($p < 0.001$).

5.2 Analysis with examples

To better understand why partial-label learning is more effective in learning the encoded knowledge, we look at cases where M_0 and M^{cd} have the incorrect segmentation while M^{pl} (and its domain adaptation variance M_+^{pl} and M_{ea}^{pl}) have the correct segmentation. We find that the majority is due to the error in re-decoded labels outside of encoded knowledge. For example, M_0 and M^{cd} give the segmentation "地震/为/**里/氏**/6.9/ 级", yet the correct segmentation given by partial-label learning is "地震/为/**里 氏**/6.9/ 级". Looking at the Wikipedia training data, there are 38 tagged entities of 里氏, but there are another 190 mentions of

[5] Statistical significance is evaluated with z-test using the standard deviation of $\sqrt{F * (1 - F)/N}$, where F is the F-measure and N is the number of words.

里氏 that are not tagged as an entity. Thus for constrained decode it sees 38 cases of "里\B 氏\E" and 190 cases of "里\S 氏\S" in the Wikipedia training data. The former comes from the encoded knowledge while the latter comes from re-decoded labels by the baseline model. The much bigger number of incorrect labels from the baseline re-decoding badly pollute the encoded knowledge. This example illustrates that constrained decode reinforces the errors from the baseline. On the other hand, the training materials for partial-label learning are purely the encoded knowledge, which is not impacted by the baseline model error. In this example, partial-label learning focuses only on the 38 cases of "里\B 氏\E" and so is able to learn that 里氏 is a word.

As a final remark, we want to make a point that, although the re-decoded labels serve to smooth out the distribution bias, the Wikipedia data is indeed not the ideal data set for such a purpose, because it itself is out of domain. The performance tends to degrade when we apply the baseline model to re-decode the out-of-domain Wikipedia data. The errorful re-decoded labels, when being used to train the model M^{cd}, could lead to further errors. For example, the baseline model M_0 is able to give the correct segmentation "电脑/**元器件**" in the CTB-6 test set. However, when it is applied to the Wikipedia data for constrained decode, for the seven occurrences of 元器件, three of which are correctly labelled as "元\B 器\I 件\E", but the other four have incorrect labels. The final model M^{cd} trained from these labels then gives incorrect segmentation "两/市/生产/的/电脑/**元/器件**/大量/销往/世界/各地" in the CTB-6 test set. On the other hand, model interpolation or EasyAdapt with partial-label learning, focusing only on the encoded knowledge and not being impacted by the errorful re-decoded labels, performs correctly in this case. For a more fair comparison between partial-label learning and constrained decode, we have also plotted the results of model interpolation and EasyAdapt for constrained decode in Figure 5. As can be seen, they improve on constrained decode a bit but still fall behind the correspondent domain adaptation approach of partial-label learning.

6 Conclusion and future work

There is rich information encoded in online web data. For example, punctuation and entity tags define some word boundaries. In this paper we show the effectiveness of partial-label learning in digesting the encoded knowledge from Wikipedia data for the task of Chinese word segmentation. Unlike approaches such as constrained decode that use the errorful re-decoded labels, partial-label learning provides a direct means to learn the encoded knowledge. By integrating some domain adaptation techniques such as EasyAdapt, we achieve an F-measure of 95.98% in the CTB-6 corpus, a significant improvement from both the supervised baseline and constrained decode. Our results also beat (Wang et al., 2011) and (Sun and Xu, 2011).

In this research we employ a sausage constraint to encode the knowledge for Chinese word segmentation. However, a sausage constraint does not reflect the legal label sequence. For example, in Figure 1 the links between label 'B' and label 'S', between 'S' and 'E', and between 'E' and 'I' are illegal, and can confuse the machine learning. In our current work we solve this issue by adding some fully-labelled data into training. Instead we can easily extend our work to use a lattice constraint by removing the illegal transitions from the sausage. The partial-label learning stands the same, by executing the forward-backward process in the constrained lattice. In future work we will examine partial-label learning with this more enforced lattice constraint in depth.

Acknowledgments

The authors would like to thank Wenbin Jiang, Xiaodong Zeng, and Weiwei Sun for helpful discussions, and the anonymous reviewers for insightful comments.

References

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Annual meetingassociation for computational linguistics*, pages 256–263. Association for Computational Linguistics.

Wenbin Jiang, Meng Sun, Yajuan Lv, Yating Yang, and Qun Liu. 2013. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, pages 761–769.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Com-*

putational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44, pages 209–216.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics*, 35:505–512.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, May.

D. C. Liu and J. Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(3):503–528, December.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 161–164, San Francisco, CA, USA.

Gideon S. Mann and Andrew McCallum. 2007. Efficient computation of entropy gradient for semi-supervised conditional random fields. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, NAACL-Short '07, pages 109–112.

Weiwei Sun and Jia Xu. 2011. Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh, Scotland, UK., July.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A discriminative latent variable Chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64.

Weiwei Sun. 2010. Word-based and character-based word segmentation models: comparison and combination. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1211–1219.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 309–317.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, pages 29–48.

Fan Yang and Paul Vozila. 2013. An empirical study of semi-supervised Chinese word segmentation using co-training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1191–1200, Seattle, Washington, USA, October. Association for Computational Linguistics.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2013a. Co-regularizing character-based and word-based models for semi-supervised chinese word segmentation. In *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, pages 171–176.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2013b. Graph-based semi-supervised model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, pages 770–779.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for Chinese word segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 311–321, Seattle, Washington, USA, October. Association for Computational Linguistics.