Conditional Random Fields with Dynamic Potentials for Chinese Named Entity Recognition

WU, Yiu Kei

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Systems Engineering and Engineering Management

Thesis/Assessment Committee

Professor Kai Pui Lam (Committee Chairman)
Professor Wai Lam (Thesis Supervisor)
Professor Jeffrey Yu (Committee Member)
Professor Zhou Shuigeng (External Examiner)

Abstract

The task of named entity recognition (NER) benefits a lot from the utilization of probabilistic frameworks recently. However, the development of Chinese NER obviously has room for improvement when comparing with English NER. The main reason is that most of the probabilistic frameworks are developed for western languages, and therefore fail to capture some specific characteristics of Chinese language effectively.

In this research work, we aim at incorporating some extensions to linear-chain conditional random fields (CRFs), which have reported the best performance on the Chinese NER task. Our approach extends linear-chain CRFs by introducing *dynamic potentials*. Dynamic potentials enable the framework to capture the dependencies across a number of states, rather than only the dependencies between adjacent states, while the inference can be kept efficient. Our experimental result shows that our proposed framework has improvement over the original CRF, which is consistent across several datasets.

摘要

近年來，命名實體識別因著概率框架的運用得到很大的助益。然而，對比英文的命名實體識別，中文命名實體識別的發展顯然還有不少改善的空間。這主要是由於絕大多數的概率框架都是爲西方語言開發的，所以它們都不能有效地捕捉漢語某些方面的特徵。

在這項研究工作中，我們的目標是對線性鏈條件隨機場的框架進行擴展。（線性鏈條件隨機場是現時在中文命名實體識別上表現最佳的概率框架。）我們的方法是將動態勢的概念引進線性鏈條件隨機場中。動態勢讓條件隨機場不單可以捕捉鄰接狀態節點的依賴關係，還可以有效率地捕捉一些橫跨多個狀態節點的依賴關係。實驗結果証明，我們提出的概率框架比原來的條件隨機場在多個數據集上有一致的改善。

Acknowledgments

There are many people I want to thank on the completion of this research work and this thesis.

First of all, I would like to express my sincere gratitude to my supervisor Dr. Wai Lam, who has given me the opportunity to pursue M.Phil. study and taste the research life. What I deeply appreciate is that he has given me not only guidance, but also lots of freedom for exploration during this research work. I have enjoyed very much in this research study of more than two years.

During my M.Phil. life, I have collaborated and communicated with quite many colleagues, and I have learned a lot from them. I'd like to thanks Gatien, Cecia, Sancho and Kaka, who have kindly helped me and given me opinions on my research. Special thanks should be given to Gatien, for he has introduced to me a lot of machine learning stuff, including CRFs.

I would like to thank my family for their understanding and support

during my research study, especially when I was very busy. Although we don't talk very much, I really know that we care about each other. Thanks to Dick, our lovely dog, who always gives us lots of fun.

Finally, I should say thanks to my dear God, Who leads me with His eternal love. He is the one that has given me the strength and faith to accomplish this research work. I know that this research work is not very great to many people, but a couple of things happened as miracles to me in this work, especially the one that happened when I was struggling about the implementation of the new algorithms. My Dear God, Thank You!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Chinese NER Problem

Named entity recognition (NER) is a useful task in the area of information extraction. NER aims at extracting and identifying all named entities (NE) precisely in text documents. The typical entity types of interest are persons, locations, and organizations. NER has been an active research area over this decade, since it helps to boost the performance for many natural language processing tasks and can be widely applied in many text-based applications.

NER benefits a lot from the development of probabilistic frameworks. In fact, probabilistic frameworks have been actively applied to English NER since about ten years ago [3, 4, 7, 20], and their current performance is shown to be satisfactory [25, 11, 19]. In contrast, the development of Chinese NER

still has room for improvement in many aspects. Therefore, we have chosen to focus this research on Chinese NER task.

Rather than developing new frameworks for Chinese NER, most research works have chosen to adapt the frameworks for English NER to Chinese NER [23, 36, 34, 17]. However, the performance of Chinese NER task is significantly lower than expected. This is mainly caused by some intrinsic properties of Chinese language, which either make Chinese NER more difficult or make the approaches for English not suitable for Chinese. The following lists these properties:

1. Lack of capitalization

   Undoubtedly, capitalization is an important clue in English NER. However, Chinese language has no analogous indicator as capitalization.

2. Lack explicit delimiters to indicate word boundaries

   Unlike the space delimiters in English text, there are no explicit delimiters to indicate word boundaries in Chinese. Human usually implicitly identifies the word boundaries based on the meaning.

3. Abbreviations

   The Chinese abbreviations are much harder to recognize, especially the organization names. In English, most abbreviations can be captured by a sequence of capital letters, e.g. IBM.

4. No "unknown words"

   Unknown words are quite strong indicators of named entities. In English, unknown words can be identified easily by matching against a lexicon. In Chinese, however, every unknown word can always be segmented to a sequence of known words.

Identifying these properties helps us to construct a suitable model for the Chinese NER task.

1.2 Contribution of Our Proposed Framework

Our proposed framework is based on conditional random fields (CRFs) [15], one of the state-of-the-art probabilistic frameworks for sequence labeling problems. The linear-chain version of CRFs reported the best performance on Chinese NER tasks in the 3rd International Chinese Language Processing Bakeoff [8, 9, 40, 32]. One of our research works, which was accepted in the 4th International Chinese Language Processing Bakeoff, also uses linear-chain CRFs as the framework and was the only group that achieved consistently high performance (higher than 90.0% in the overall F-measure) on all the corpora in the open track of the Chinese NER task in the 6th SIGHAN Workshop [35]. Therefore, the framework we propose in this thesis aims at preserving the powerful inference of linear-chain CRFs, as well as im-

porting the ability to capture some useful long-range dependencies among states, which is similar to that in the general-graph version of CRFs [31]. Although general-graph CRFs can also capture the Markov dependencies as in linear-chain CRFs, the inference it offers can only be an approximation and inefficient. Therefore, rather than resorting to general-graph CRFs, we introduce *dynamic potentials* to linear-chain CRFs.

In our proposed framework, the desired long-range state dependencies are captured by dynamic potentials on a linear-chain structure, rather than using the potentials on some fixed edges in a graphical structure. By modifying the common Viterbi procedure and forward-backward procedure for linear-chain CRFs, the effectiveness and the efficiency of the exact inference can be kept. In practice, the extra time required by the new inference algorithm is at most 60% of the time required by the original one in linear-chain CRFs.

By means of dynamic potentials, it is much more easy and effective to embed human knowledge about named entities, e.g. name structures, grammar rules, etc., in the model as a form of features. This is usually not possible in original linear-chain CRFs. This is especially true for Chinese NER, because a Chinese named entity usually occupies no less than 3 states, and the range of state dependencies in linear-chain CRFs cannot be more than 2 states in order to keep its inference tractable.

Another probabilistic framework called semi-Markov CRFs (semi-CRFs)

[27] shares some similarities with our proposed framework, mainly on the ability to capture those long-range state dependencies. However, due to the differences between the formulations of the two frameworks, our proposed framework does not suffer from the problems found in semi-CRFs on NER tasks, including hard entity length limit and search space problem. In addition, our model can also capture the dependencies of two entities which are separated by a sequence of "out-of-entity" states while semi-CRFs cannot. These issues will be discussed in detail in Chapter 3.

Our experimental result shows that our framework has improvement over the original CRFs. Such improvement is consistent across several datasets. The details will be presented in Chapter 4.

Chapter 2

Related Work

A variety of probabilistic frameworks have been applied to Chinese NER tasks, including the hidden Markov model (HMM), the maximum entropy model (MaxEnt), the conditional random field (CRF), etc. In this chapter, some of these research models will be reported. The limitations of these probabilistic frameworks will also be discussed.

In most of these research models, some techniques were developed and applied upon the probabilistic frameworks, e.g. heuristics, preliminary word segmentation, separate models for different NE, etc. Although these techniques sometimes help to boost the performance, they will not be discussed in depth in this thesis, as the aim of this research is to improve the underlying probabilistic framework. Nevertheless, it should not be difficult to apply these techniques on the improved framework to boost the performance

further.

2.1 Hidden Markov Models

The hidden Markov model (HMM) is the first statistical framework that was successfully applied to NER tasks [4]. Many Chinese NER researches used HMM as the underlying framework and achieved promising performance [33, 29, 37, 12].

The reason why the HMM approach can achieve good performance on NER is its capability to capture the most important information to recognize a name, including the local context and the internal information of a name, as well as the linear structure of the sentence.

However, HMMs also suffer from a number of limitations:

1. Generative modeling

   An HMM is a generative model, which means it learns how to generate the observation sequence. The consequence is that an HMM is required to enumerate the whole space of observation sequences, such to guarantee all possible sequence can be generated by the model.

2. Inability to use rich representation of observations

   As a result of generative modeling, rich representation of observations

can make the inference intractable. Therefore, besides Markov assumption, additional independence assumptions are enforced among the observation tokens within a sequence. The reality is that, within an observation sequence, there exists a number of features which give very useful information when they are used together. In practice, ignoring inter-dependent features and making such independence assumptions hurt the performance a lot.

2.2 Maximum Entropy Models

The maximum entropy model (MaxEnt) [2] is very popular among English NER researches [6, 5, 10], but this is not the case for Chinese NER. Nevertheless, applying MaxEnts to Chinese NER has been investigated [13, 39], and they show that MaxEnts generally outperforms HMMs.

The MaxEnt is a probabilistic framework that is based on the principle of maximum entropy in the field of information theory. The main idea of the MaxEnt is that, given some known statistics information about a classification task, the objective is to maximize the entropy (i.e. the uncertainty about correct classification) such that the assumption is minimized and therefore the parameter estimation would not be biased.

Unlike HMMs, an MaxEnt is a discriminative model, i.e. an MaxEnt

learns how to discriminate, rather than generate, an observation sequence [21]. For general classification task, as reported by [22], it is theoretically proved that discriminative models converge to the limit of linear classifiers when given enough training data, and therefore should not be worse than generative models. In practice, the performance of discriminative models is usually better than generative models.

Discriminative modeling does not require the model to enumerate the whole observation space. Therefore, a discriminative model is able to use much richer representation of observations and allows exploiting inter-dependent features within an observation sequence simultaneously.

One major drawback of the MaxEnt is that, it has to give up either the probabilistic state transition or the optimal state sequence searching algorithm. Chieu and Ng chose to have deterministic state transitions for optimal state sequences [10], while Ratnaparkhi et al. chose to use beam search for third Markov order probabilistic state transitions [24]. Obviously, both are desired components for the framework. Giving up either one of them just significantly hurts the performance.

In addition, the MaxEnt suffers from the label bias problem [15]. This problem is common to all discriminative models which are trained by per-state normalization. Such normalization makes these models bias to choose the paths that pass through the states with fewer outgoing transitions or with

lower entropy next state distribution in general, because the probabilities of the paths getting through these states are always estimated higher.

2.3 Conditional Random Fields

The conditional random field (CRF) [15] is a sequence labeling framework that aims at bringing together the power of the HMM and the MaxEnt, as well as dealing with the label bias problem. As it has inherited the strengths and overcome the limitations of the previous frameworks, it generally performs the best on sequence labeling task.

Currently, the CRF is the most active framework that is being applied to Chinese NER task. In the 3rd International Chinese Language Processing Bakeoff of SIGHAN [16], almost all top-ranked Chinese NER systems used CRFs as the basic framework [8, 9, 40, 32]. This competition was divided into close track and open track. Close track does not allow to use any external resources, while open track does allow.

Among these research models, Chen and Shan et al. used only the CRF of the simplest form with basic features [8] to achieve comparatively good performance in close track, which was also consistent across a few data sets [16]. In their paper, it is shown that their CRF models significantly outperformed their MaxEnt model, even though they used much more features

for the MaxEnt model in order to capture more information. They also show that heuristic post-processing to CRFs output may degrade the overall performance.

Similar to the previous group, Chen and Zhang et al. participated in close track and used the CRF of the simplest form as the framework, but they implemented a much richer set of feature functions for their CRF model [9]. However, this did not help them to achieve a significantly higher performance than the previous group. This group also implemented a post-processing algorithm to correct the inconsistent tags by utilizing the top 20 output sequences of each sentence. The general idea of the algorithm is to accept the named entities in the output with high confidence score and use them to correct those output with low confidence score. This algorithm consistently raised the performance, but not significantly.

Zhou et al. constructed a multi-phase model by arranging a few CRF models in cascading manner, i.e. the output of a CRF model would be the input of the next CRF model [40]. The first layer is a character-level CRF for word segmentation. The next layer is a word-level CRF for labeling person names, followed by the word-level CRF models for location name and organization names. In close track, this construction gave an insignificant gain in F-score over the simplest CRF model on one dataset. In open track, this system successfully achieved the best performance among those systems

which used CRFs as framework, but it still significantly lagged behind the one achieving the first place, which used the MaxEnt as the basic framework [38]. This indicates that there should be room for improvement in incorporating external knowledge to CRFs.

To conclude, the CRF significantly outperforms other probabilistic frameworks in learning statistical information from a corpus, while it may not be as good in incorporating external knowledge. The current ways to improve the performance can be generalized into three categories, i.e. using richer feature set in CRFs, preprocessing (e.g. preliminary word segmentation), and post-processing. Although these techniques helps to improve performance when external knowledge is supplied, they in fact do not help CRFs much to learn more information *from the corpus*.

In this research, we choose to improve the performance of Chinese NER by overcoming the limitations of the underlying probabilistic framework. The following lists the limitations of the CRF:

1. Inability to incorporate some external knowledge

   As discussed before, the CRF is not as good as other probabilistic frameworks in this aspect. Much knowledge cannot be naturally represented as features in CRFs, e.g. structures of names. The most natural way to define feature functions in CRFs for Chinese NER is to join local

tokens together. Such features are not easy to understand for human.

2. Inability to capture long-range dependencies between states

   Although the CRF has already solved the long-range dependencies between tokens, there exists useful long-range dependencies not captured by the framework. An extreme example is reported by Chen and Shan et al. in [8]. They found that their CRF models failed to recognize many person names in a name list that appeared in a sentence. In such situation, it would be valuable if the long-range dependencies between entities, which are across the separating punctuations or words, can be captured.

The details of these limitations and their solutions will be presented in the next chapter.

Chapter 3

Our Proposed Model

3.1 Background

3.1.1 Problem Formulation

Chinese NER problem can be formulated as a sequence labeling problem. In such formulation, a Chinese sentence is treated as a sequence of tokens and each Chinese character is treated as a token. The objective is to decide the appropriate label to each token.

However, since an entity may consist of two or more tokens, the entity labels have to be designed in such a way that the boundaries of the entities can be identified. Different labeling schemes have been investigated. The most basic scheme is to add "B" and "I" to each entity label to form entity

tags. The B-tag of entity type ξ (denoted by ξ_B) is to tag the beginning token of an entity of type ξ, while the I-tag (denoted by ξ_I) is to tag the remaining tokens inside the entity. O is to label the tokens which are not any part of the entities. This scheme is know as BIO. Another basic scheme is to add "I" and "E" to each entity label to form entity tags instead. The functions of I-tags and O are the same as in the previous scheme, while the E-tag of entity type ξ (denoted by ξ_E) is to tag the ending token of an ξ entity. This scheme is known as IOE. These two schemes are actually capturing slightly different information. The former scheme is better in capturing the front boundary of the entity, while the latter is better in capturing the rear boundary. And they seldom perform significantly better than the other. A more complicated scheme is to add "B", "I", "E" and "S" to each entity label to form entity tags. The B-, I-, E-tags and O have the same functions as in previous labeling schemes. The additional S-tag for entity type ξ (denoted by ξ_S) is to label those ξ entities which consist of only one single character. We call this scheme BIOES. This scheme generally leads to a better performance than BIO and IOE, since it provides more information for the probabilistic framework to capture.

Depending on the probabilistic framework, it may be additionally required that the document should be segmented by sentences rather than by paragraphs. Such requirement helps the framework to estimate the param-

eters more accurately. The conditional random field (CRF) is one of these frameworks. And since our proposed framework is based on the CRF, it also has this requirement.

3.1.2 Conditional Random Fields

In this section, the linear-chain version of the CRF will be briefly introduced. The notations for the CRF will be adapted from [31] and [28]. This will serve as the basis for developing the notations for our proposed framework. The limitations of the CRF will also be discussed in detail.

A linear-chain CRF defines the conditional probability distribution of label sequences given observation sequences. The label sequence is also known as the output sequence of the CRF. In the following, $\boldsymbol{y}$ and $\boldsymbol{x}$ are used to represent the output sequence and the observation sequence respectively, while y_i and x_i are used to represent the single output and observation variables at position i respectively. Note that an observation in CRFs is not necessarily a token, but can also be a vector of features.

The term "linear-chain" refers to the shape of the graph formed by the output variables, i.e. the output variables line up in order and link with the adjacent ones to form a chain. To illustrate it with a diagram, we adopt the representation from [31], and it is shown in Figure 3.1.

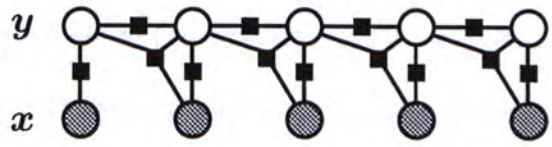

Figure 3.1: Graphical illustration of a typical linear-chain CRF.

In Figure 3.1, each unshaded node represents an output random variable while each shaded node represents an observation. Each edge represents a dependency relationship, while each small black square on an edge represents the corresponding potential function. We use Ψ to denote a potential. Regardless of the observation variables, there are two basic types of potentials: $\Psi(i, y_i)$ denotes the potential occurring at position i which depends only on y_i, and $\Psi(i, y_{i-1} \rightarrow y_i)$ denotes the potential occurring at position i which depends on *both* y_{i-1} and y_i. $\Psi(i, y_{i-1}y_i)$ is the composite potential that represents the product of the two potentials. Note that these potentials are also dependent on the corresponding observation variables. However, since the observations variables do not matter much in this research, they are omitted here to keep the formulation simpler. Figure 3.1 can then be simplified to Figure 3.2.

Figure 3.2: Simplified graphical illustration of a typical linear-chain CRF.

The value of a potential function depends on the corresponding feature

functions and their weights. There are two types of features corresponding to the two basic types of potentials. s_k denotes the k^{th} state feature and $s_k(i, y_i, x_i)$ gives its value for the potential $\Psi(i, y_i)$. t_k denotes the k^{th} transition feature and $t_k(i, y_{i-1} \rightarrow y_i, x_i)$ gives its value for the potential $\Psi(i, y_{i-1} \rightarrow y_i)$. Each feature has a weight that is denoted by λ. The values of the potentials are given by the following formulae:

$$\Psi(i, y_i) = \exp\left\{\sum_k \lambda_{s_k} s_k(i, y_i, x_i)\right\} \text{ , where } i \geq 1, \tag{3.1}$$

and

$$\Psi(i, y_{i-1} \rightarrow y_i) = \exp\left\{\sum_k \lambda_{t_k} t_k(i, y_{i-1} \rightarrow y_i, x_i)\right\} \text{ , where } i > 1. \tag{3.2}$$

The composite potential is defined as:

$$\Psi(i, y_{i-1} y_i) = \begin{cases} \Psi(i, y_{i-1} \rightarrow y_i)\Psi(i, y_i) & i > 1 \\ \Psi(i, y_i) & i = 1 \end{cases} \tag{3.3}$$

To make the notation of feature functions more uniform, we also use the following to represent the original ones:

$$\begin{aligned} s_k(i, \boldsymbol{y}, \boldsymbol{x}) &= s_k(i, y_i, x_i) \\ t_k(i, \boldsymbol{y}, \boldsymbol{x}) &= \begin{cases} t_k(i, y_{i-1} \rightarrow y_i, x_i) & i > 1 \\ 0 & i = 1 \end{cases} \end{aligned}$$

And we use the feature function vector $\boldsymbol{f}(i, \boldsymbol{y}, \boldsymbol{x})$ to group all these state features and transition features together, and it gives the vector of values

outputted by these feature functions for position i. We also use the weight vector $\boldsymbol{\lambda}$ to group the weights for these feature functions. Therefore,

$$\Psi(i, y_{i-1}y_i) = e^{\boldsymbol{\lambda}\cdot\boldsymbol{f}(i,\boldsymbol{y},\boldsymbol{x})}. \tag{3.4}$$

Since a CRF normalizes according to the whole sequence, there is a global feature vector that accounts for all the feature function vectors in the sequence:

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \boldsymbol{f}(i, \boldsymbol{y}, \boldsymbol{x})$$

where i ranges over the possible positions of the sequence.

Then, the conditional probability distribution that a CRF defines can be expressed as:

$$p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z_{\boldsymbol{\lambda}}(\boldsymbol{x})} e^{\boldsymbol{\lambda}\cdot\boldsymbol{F}(\boldsymbol{y},\boldsymbol{x})} \tag{3.5}$$

where

$$Z_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} e^{\boldsymbol{\lambda}\cdot\boldsymbol{F}(\boldsymbol{y},\boldsymbol{x})}$$

The most probable label sequence for input sequence $\boldsymbol{x}$ is

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y}} p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) \\
&= \operatorname*{argmax}_{\boldsymbol{y}} e^{\boldsymbol{\lambda}\cdot\boldsymbol{F}(\boldsymbol{y},\boldsymbol{x})} \\
&= \operatorname*{argmax}_{\boldsymbol{y}} \prod_i e^{\boldsymbol{\lambda}\cdot\boldsymbol{f}(i,\boldsymbol{y},\boldsymbol{x})} \\
&= \operatorname*{argmax}_{\boldsymbol{y}} \prod_i \Psi(i, y_{i-1}y_i)
\end{aligned}$$

Let $\delta(i, y)$ be the maximum likelihood among all the output sequences that end at position i with output y. Note that $\boldsymbol{x}$, $\boldsymbol{f}$ and $\boldsymbol{\lambda}$ are omitted here for simplicity, although the value of $\delta(i, y)$ also depends on them. The famous Viterbi algorithm for CRFs can then be implemented using the following recursion:

$$\delta(i, y) = \begin{cases} \max_{y'} \{\delta(i-1, y') \cdot \Psi(i, y'y)\} & i > 0 \\ 1 & i = 0 \end{cases} \tag{3.6}$$

The maximum likelihood is given by $\max_y \delta(|\boldsymbol{x}|, y)$, and its corresponding path is the best output sequence for $\boldsymbol{x}$.

Given a training set $T = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^N$, a CRF model is trained by maximizing the log-likelihood $\mathcal{Q}_{\boldsymbol{\lambda}}$:

$$\mathcal{Q}_{\boldsymbol{\lambda}} = \sum_k \log p_{\boldsymbol{\lambda}}(\boldsymbol{y}_k | \boldsymbol{x}_k) \tag{3.7}$$

$$= \sum_k (\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - \log Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k)) \tag{3.8}$$

For the convexity of Eq. (3.8), $\mathcal{Q}_{\boldsymbol{\lambda}}$ is optimized when the gradient of $\mathcal{Q}_{\boldsymbol{\lambda}}$ is zero.

$$\begin{aligned} \nabla \mathcal{Q}_{\boldsymbol{\lambda}} &= \sum_k \left[\boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - \sum_{\boldsymbol{y}'} \boldsymbol{F}(\boldsymbol{y}', \boldsymbol{x}_k) \cdot e^{\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}', \boldsymbol{x}_k)} \cdot \frac{1}{Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k)} \right] \\ &= \sum_k \left[\boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - \sum_{\boldsymbol{y}'} \boldsymbol{F}(\boldsymbol{y}', \boldsymbol{x}_k) \cdot p_{\boldsymbol{\lambda}}(\boldsymbol{y}' | \boldsymbol{x}_k) \right] \\ &= \sum_k \left[\boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - E_{p_{\boldsymbol{\lambda}}(\boldsymbol{Y} | \boldsymbol{x}_k)} \boldsymbol{F}(\boldsymbol{Y}, \boldsymbol{x}_k) \right] \end{aligned}$$

The interpretation of this equation is that, when the expected value of $\boldsymbol{F}$ under the empirical distribution equals to the expectation of $\boldsymbol{F}$ under the model distribution, the maximum of $\mathcal{Q}_{\boldsymbol{\lambda}}$ is reached.

The model expectation can be computed using the forward-backward algorithm of the CRF. First, let the forward variable $\alpha(i, y)$ be the sum of likelihood scores of all the output sequences that end at position i with output y. We can use the following recursion to calculate each α:

$$\alpha(i, y) = \begin{cases} \sum_{y'} \{\alpha(i-1, y') \cdot \Psi(i, y'y)\} & 0 < i \leq |\boldsymbol{x}| \\ 1 & i = 0 \end{cases} \tag{3.9}$$

The normalization factor $Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k)$ can be obtained using the forward variables:

$$Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k) = \sum_{y'} \alpha(|\boldsymbol{x}|, y') \tag{3.10}$$

Then, let the backward variable $\beta(i, y)$ be the sum of likelihood scores of all the labelings from the end of the sequence to position $(i+1)$ with output y at position i. We can use the following recursion to calculate each β:

$$\beta(i, y) = \begin{cases} \sum_{y'} \{\Psi(i+1, y'y) \cdot \beta(i+1, y')\} & 0 \leq i < |\boldsymbol{x}| \\ 1 & i = |\boldsymbol{x}| \end{cases} \tag{3.11}$$

To compute the model expectation of the feature functions, we need to compute the marginal distributions. This can be done by utilizing the for-

ward and backward variables:

$$p(y_{i-1}, y_i|\boldsymbol{x}) = \alpha(i-1, y_{i-1})\Psi(i, y_{i-1}y_i)\beta(i, y_i) \cdot \frac{1}{Z_{\boldsymbol{\lambda}}(\boldsymbol{x})} \tag{3.12}$$

After describing the whole framework of the CRF, we explain its limitations in details. The first limitation is its inability to incorporate some external knowledge for Chinese NER.

As discussed in Chapter 2, the most natural and effective way to define feature functions in CRFs for Chinese NER is to use the local tokens and to join them together, which is supported by [8]. This research work also found that, when determining the output for position i, the most helpful information would be from the characters from position $(i-2)$ to $(i+2)$, namely C_{i-2}, C_{i-1}, C_i, C_{i+1} and C_{i+2}. We use the sequence shown in Figure 3.3 as an example.

position	1	2	3	4	5	6	7	8	9	10	
token	香	港	工	程	師	學	會	即	將	假	
output	Org_B	Org_I	Org_I	Org_I	Org_I	Org_I	Org_E	O	O	O	
(cont.)	11	12	13	14	15	16	17	18	19	20	21
	香	港	會	展	新	翼	舉	辦	講	座	。
	Loc_B	Loc_I	Loc_I	Loc_I	Loc_I	Loc_E	O	O	O	O	O

Figure 3.3: An example sentence with the correct labeling

The features activated to recognize the first entity in the example sentence would be like those shown in Figure 3.4.

position	transition features	state features
i=1	Start→Org$_B$	C_i='香', C_{i+1}='港', C_{i+2}='工', C_iC_{i+1}='香港', $C_{i+1}C_{i+2}$='港工'
i=2	Org$_B$→Org$_I$	C_{i-1}='香', C_i='港', C_{i+1}='工', C_{i+2}='程', $C_{i-1}C_i$='香港', C_iC_{i+1}='港工", $C_{i+1}C_{i+2}$='工程'
i=3	Org$_I$→Org$_I$	C_{i-2}='香', C_{i-1}='港', C_i='工', C_{i+1}='程', C_{i+2}='師', $C_{i-2}C_{i-1}$='香港', $C_{i-1}C_i$='港工", C_iC_{i+1}='工程', $C_{i+1}C_{i+2}$='程師'
i=4	Org$_I$→Org$_I$	C_{i-2}='港', C_{i-1}='工', C_i='程', C_{i+1}='師', C_{i+2}='學', $C_{i-2}C_{i-1}$='港工', $C_{i-1}C_i$='工程', C_iC_{i+1}='程師', $C_{i+1}C_{i+2}$='師學'
i=5	Org$_I$→Org$_I$	C_{i-2}='工', C_{i-1}='程', C_i='師', C_{i+1}='學', C_{i+2}='會', $C_{i-2}C_{i-1}$='工程', $C_{i-1}C_i$='程師', C_iC_{i+1}='師學', $C_{i+1}C_{i+2}$='學會'
i=6	Org$_I$→Org$_I$	C_{i-2}='程', C_{i-1}='師', C_i='學', C_{i+1}='會', C_{i+2}='即', $C_{i-2}C_{i-1}$='程師', $C_{i-1}C_i$='師學', C_iC_{i+1}='學會', $C_{i+1}C_{i+2}$='會即'
i=7	Org$_I$→Org$_E$	C_{i-2}='師', C_{i-1}='學', C_i='會', C_{i+1}='即', C_{i+2}='將', $C_{i-2}C_{i-1}$='師學', $C_{i-1}C_i$='學會', C_iC_{i+1}='會即', $C_{i+1}C_{i+2}$='即將'

Figure 3.4: Features activated for recognizing the first entity name in the example sentence

Although it has been shown that these simple features help to achieve consistently good performance across different datasets, it is also obvious that such features are difficult to comprehend for human, as we do not recognize the names in a sentence like this. This brings difficulties in further analysis about the learnt model. And more importantly, human knowledge can hardly be incorporated. For example, name structures often help us to recognize entity names. "[city_name][profession_title]學會" is one of the general structures that help us to recognize "香港工程師學會" in the example

sentence, as well as other similar names such as "北京律師學會".

These kinds of knowledge are very helpful, but they cannot be naturally expressed as features in CRFs. One may argue that we can have some feature that activates at the end of the name when the current and previous characters match the structure. However, such strategy would add scores to all labelings whichever has an Org_F tag for position 7. This would adversely affects the fairness among the labelings, since the labelings which deserve this score should be only those having the positions 1 to 7 labeled as a whole organization name.

Another limitation of CRF is the inability to capture long-range dependencies between states. Using the example sentence again, it helps if the dependency between the organization and the location with the phrase "即將假" in-between can be captured. Another example is reported by Chen and Shan et al. in [8]. They found that their CRF models fail to recognize many person names in a name list that appeared in a sentence. The situation can be illustrated using the sequence shown in Figure 3.5.

Consider the second person name "談世中" and its context, as well as their true outputs. The name itself does not provide strong evidence to show it is a person name. In such situation, the context plays an important role. Unfortunately, under the formulation of the CRF, the help from the context is also very limited in such situation. The punctuations "、" on each side of

position	1	2	3	4	5	6	7	8	9	10	
token	學	者	專	家	谷	源	洋	、	談	世	
output	O	O	O	O	Per_B	Per_I	Per_E	O	Per_B	Per_I	
(cont.)	11	12	13	14	15	16	17	18	19	20	⋯
	中	、	陳	漓	高	、	甄	炳	禧	等	⋯
	Per_E	O	Per_B	Per_I	Per_E	O	Per_B	Per_I	Per_E	O	⋯

Figure 3.5: Another example sentence with the correct labeling

the name may be able to indicate that it is in a list, but cannot show this is a list of person names. The first character of the third name, i.e. 陳, possibly provides some hints as it is one of the most common Chinese surname, but itself would be too weak as it also frequently appears in the words that are not person names.

Apart from the limitation of CRFs, let us consider other person names in the sentence. The first name should be recognized with a much higher probability, as the preceding phrase "專家" provides strong evidence and the surname "谷" also gives a little help. The fourth name should also be recognized with a high probability, as the surname gives strong evidence and the remaining characters also appear quite frequently in given names. The third name should not be too difficult, as its first character is a common Chinese surname. Therefore, the whole scenario would be different if the other names are known to have high probabilities of being labeled as person names when the outputs of the tokens in the second name are being deter-

mined. This can be done if the dependencies between the names across the punctuation " 、 " can be captured. In this way, the high probabilistic scores of some names in the list can be shared to the names with low scores.

Using the original formulation of the CRF, the only possible way to capture these long-range dependencies between states is by the general-graph version of the CRF. The disadvantages of this approach are already discussed before. One more limitation is that, correct dependency edges have to be added to the graph in prior, which is obviously not always feasible [30]. Therefore, we will not discuss the general-graph CRF in depth here, but just show the diagram of a general-graph CRF model in Figure 3.6.

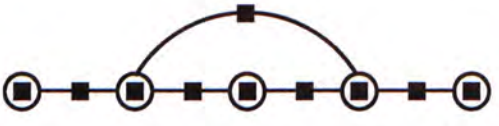

Figure 3.6: Graphical illustration of a general-graph CRF.

3.1.3 Semi-Markov Conditional Random Fields

As mentioned in Chapter 1, another framework called the semi-Markov CRF (semi-CRF) shares some similarities with our proposed framework. However, semi-CRF suffers from a number of limitations as explained below.

A semi-CRF is a model that is trained based on the principle of CRFs. In contrast to CRFs, a semi-CRF models the data using semi-Markov chains

rather than Markov chains. In effect, a semi-CRF takes a sequence of tokens as input, just like CRFs, but it outputs a sequence of labeled segments, where a segment may contain more than one token. Because of this formulation, there is no need to add any "B", "I", "E" or "S" to the entity labels in order to indicate entity boundaries for NER tasks, as an entity can be represented by a segment. The formulation has a variable L which defines the maximum number of tokens allowed in a segment. The formulation and the algorithm do not impose any hard restriction on L, but as the time complexity grows linearly with it, L should be small enough in practice, usually not bigger than 10. For Chinese NER tasks, such restriction is not practical. Using People's daily (January 1998) corpus, one of the corpora we used for our experiments, as an example, even setting L to be 23 cannot cover all the entities in the corpus. In such kind of situations, the sentences with any entity longer than L tokens cannot be used in training, and any entity longer than L tokens definitely cannot be recognized in testing.

In a later version of the semi-CRF [26], the restriction on segment length is removed. However, this in turn introduces another problem on the search space. Consider a sequence of n tokens and count only the labelings with all tokens labeled as "O", i.e. the "out-of-entity" label. For an original CRF, there is always only one such labeling. For a semi-CRF, in contrast, there are always at least two different labelings as long as n is bigger than

1. Note that a semi-CRF also distinguishes different segmentations, e.g. it regards "[O] [O]" and "[OO]" as two different labelings. And it should also be noted that the number of these different labelings grows rapidly with the number of consecutive O labels in an original labeling. This not only wastes resources for searching among the labelings that are not of any interest, it also interferes the parameter estimation in training, as parameters are unnecessarily estimated for different segmentation of consecutive O labels.

Theoretically, it is proved that the semi-CRF is strictly more expressive than the CRF [1], i.e. the feature functions defined in a CRF can be somehow defined in a semi-CRF, but not vice versa. Nevertheless, semi-CRFs cannot always out-perform CRFs in practice. On the other hand, CRFs often has better performance when enough data is given [1, 18].

3.2 The Formulation of Our Proposed Model

3.2.1 The Main Principle

Our proposed framework is formulated to effectively inherit the strength of the inference of linear-chain CRFs, and efficiently capture some useful long-range state dependencies. In this section, the modeling and the main principle of the algorithms in our proposed framework will be presented. The

detailed derivation of the formulation will be given in the next section.

The modeling of our proposed framework can be illustrated by the graph shown in Figure 3.7. The main characteristic that makes this graphical representation different from that of linear-chain CRF (i.e. Figure 3.2) is the edge connecting two states with some other states in-between. This representation shares some resemblances to the general-graph CRF, as shown in Figure 3.6. However, the potential on the long-range dependency edge in this diagram is represented by a triangle. This triangle indicates that this potential is different from the usual one. We call such kind of potentials as *dynamic potentials*. And from now on, we will refer our proposed framework as "CRFDP", i.e. conditional random fields with dynamic potentials.

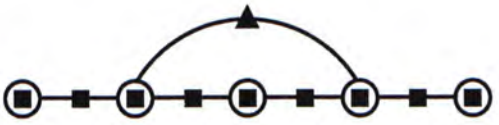

Figure 3.7: Graphical illustration of our proposed framework (i.e. CRFDP).

The term "dynamic" is used to denote the fact that such potentials are not static, as their existence depends on some conditions. In other words, a dynamic potential exists for some labelings for a sequence of tokens, but not for some other labelings for the same sequence. In order to achieve efficient inference, we have to impose some restrictions on these conditions. At the same time, we should also capture the useful long-range dependencies

as many as possible. To achieve good balance, our proposed framework is formulated to capture two most important kinds of long-range dependencies: (1) the dependencies linking the ending state of an entity and the preceding state of the entity, and (2) the dependencies linking two entities with a number of O-states in-between. The first kind of dependencies helps more advanced and more accurate modeling within the entity, while the second kind helps to model the relationship between two entities, even when they are separated by some non-entity tokens.

To capture these dependencies, it is obviously required to infer across a number of states at a time, which is computationally infeasible in the original linear-chain CRF. By introducing dynamic potentials, it is however possible to develop efficient inference algorithms based on the ones for the original CRF.

Note that calculating marginal distributions is required in model training, while searching for the optimal labeling is required in model application. Both tasks require to effectively enumerate all the possible labelings for a token sequence. With the first or second Markov order assumption in the original CRF, dynamic programming can be applied to obtain computation results efficiently as if all labelings are enumerated. Using the same principle, the forward-backward algorithm is developed to accomplish the former task, while the Viterbi algorithm is developed to accomplish the latter task. For

the modeling in our proposed framework, however, the Markov assumption does not necessarily exist. Therefore, the Viterbi and forward-backward algorithms have to be modified to serve the same objectives efficiently. Since both algorithms use the same principle to achieve efficient computation, we will here explain only the principle. The derivation of both algorithms will be presented in the next section.

To explain the principle more clearly, we will make use of the following graphical notations. (Note: The X in the following notations has the meaning of "all possible labelings".)

1. $\mathrm{X}\boxed{_j\xi_i}$ represents the set of all possible labelings up to position i, with an entity of type ξ which ends at i and begins at j.

2. $\mathrm{X}\overline{_j\xi_i}$ represents the set of all possible labelings up to position i, with an entity of type ξ which ends at i and begins before j but not at j.

3. $\mathrm{X}\overline{\boxed{_j\xi_i}}$ represents the set of all possible labelings up to position i, with an entity of type ξ which ends at i and begins before or at j. Therefore, this set is the union of the above two sets.

Suppose we want to know some calculation result about all possible labelings up to position i which have an entity of type ξ ending at position i, i.e. $\mathrm{X}\overline{\boxed{_i\xi_i}}$. In order to solve this problem efficiently, repeated calculation should be avoided. This can be achieved by decomposing the problem to

some sub-problems, which can be solved efficiently by reusing some results calculated from previous positions.

Note that the labeling set $X_{i\,\boxed{\xi_i}}$ can be decomposed into two disjoint labeling sets: $X_{\boxed{i\,\xi_i}}$ and $X_{i\text{-}1\,\boxed{\xi_i}}$. Therefore, the problem of $X_{i\,\boxed{\xi_i}}$ can be solved by combining the solutions for the problems of these two labeling sets. The problem of $X_{i\text{-}1\,\boxed{\xi_i}}$, in turn, can be similarly decomposed to the problems for $X_{\boxed{i\text{-}1\,\xi_i}}$ and $X_{i\text{-}2\,\boxed{\xi_i}}$. This decomposition process can be done recursively. However, in order to avoid such recursive process always looping all the way back to the beginning of the token sequence, which is time-wasting for long sentences and usually not necessary, we choose to set some upper limit for the number of steps going backwards. Note that such limit needs not to be the same for all entity types, and it is also beneficial to set different limits for different entity types, as some types of entity tend to be long (e.g. organization names) while some tend to be short (e.g. person names). This measure effectively restricts the entity candidates that are longer than their corresponding limit from using dynamic potentials, and therefore no need to pay for it. In practice, these limits can be set to quite long to cover most entities (more than 99%) while the inference can still be kept efficient.

Let m_ξ be the maximum length of the entities of type ξ which are allowed to use dynamic potentials. Then, the original problem of $X_{i\,\boxed{\xi_i}}$ can eventually be decomposed to the problems of $X_{\boxed{j\,\xi_i}}$ for $j = i, (i-1), ..., (i-m_\xi+1)$

and the problem of $\mathrm{X}_{\boxed{i\text{-}m_\xi \xi_i}}$.

Each of these sub-problems, except the last one, can efficiently be solved by reusing the solutions of each kind of labels for some previous position and at the same time accounting for the last appended ξ entity. (Note that for the case that the previous label is O, the dependencies between the newly appended entity and its previous entity have to be considered. This issue will be dealt shortly later. At this moment, let us assume that this problem is solved.)

For the last sub-problem, i.e. the one for $\mathrm{X}_{\boxed{i\text{-}m_\xi \xi_i}}$, we should observe that the length of the last ξ entity exceeds the limit m_ξ, and so dynamic potentials would not be applied to the entity. Then, this sub-problem can be solved efficiently if we additionally keep the result calculated for the set of all possible labelings up to $(i - m_\xi)$, which the tokens from $(i - m_\xi)$ to i would be labeled as *one* ξ entity, i.e. the ξ entity begins at or before $(i - m_\xi)$ and ends at or after i. Then, by reusing this result and together accounting for the ξ entity that ends at i, the sub-problem for $\mathrm{X}_{\boxed{i\text{-}m_\xi \xi_i}}$ can be solved easily. The additionally kept result can be reused to compute the one of the same kind for position $(i + 1)$. This is achieved by taking into account the result calculated up to position $(i - m_\xi)$ for the set of labelings that a ξ entity has begun at position $(i - m_\xi)$ and will be longer than m_ξ tokens

(so that dynamic potentials need not to be considered for this entity). This calculation is simple and will be shown later in the detailed formulation.

The above description explains the idea about how the long-range dependencies linking across an entity candidate can be inferred efficiently. A similar idea can be applied to efficiently account for the long-range dependencies linking two entity candidates with a number of O-labeled tokens in-between. To illustrate the idea, the following notations will be used this time:

1. $\mathrm{X}\xi'\boxed{_j\mathrm{O}_i}\xi''$ represents the set of all possible labelings up to position i, with a sequence of O-labeled tokens starting at (but not before) j and ending at i, which follows an entity of type ξ' and to be followed by an entity of type ξ''.

2. $\mathrm{X}\boxed{_j\mathrm{O}_i}\xi''$ represents the set of all possible labelings up to position i, with a sequence of O-labeled tokens starting strictly before j and ending at i, which is to be followed by an entity of type ξ''.

3. $\mathrm{X}\boxed{\vdots_j\mathrm{O}_i}\xi''$ represents the set of all possible labelings up to position i, with a sequence of O-labeled tokens starting at or before j and ending at i, which is to be followed by an entity of type ξ''.

Suppose we want to know some calculation results about the labeling set $\mathrm{X}\boxed{\vdots_i\mathrm{O}_i}\xi''$ (Note: this problem is the one we assumed to be solved previously).

This labeling set can be decomposed into a few disjoint labeling sets, i.e. $\mathrm{X}\xi'\boxed{_i\mathrm{O}_i}\xi''$ for each entity type ξ' and $\mathrm{X}\boxed{_{i\text{-}1}\mathrm{O}_i}\xi''$. The last labeling set can be in turn decomposed into a few labeling sets similarly. This decomposition process can be done recursively, but we also set a limit for this such that the inference can be kept efficient.

Let m_O be the maximum length of the O-labeled token sequence between two entities where dynamic potential is allowed to be applied to model the relationship between the entities. Then, the original problem of $\mathrm{X}\boxed{_i\mathrm{O}_i}\xi''$ can be eventually decomposed to the problems of $\mathrm{X}\xi'\boxed{_j\mathrm{O}_i}\xi''$ for each entity type ξ', where $j = i, (i-1), ..., (i-m_O+1)$, and then the problem of $\mathrm{X}\boxed{_{i\text{-}m_O}\mathrm{O}_i}\xi''$.

Except the last sub-problem, each of these sub-problems, say the problem of $\mathrm{X}\xi'\boxed{_j\mathrm{O}_i}\xi''$, can be solved efficiently by reusing the solutions for the corresponding previous position with an ξ' entity appended (i.e. the solution of $\mathrm{X}\boxed{_{j\text{-}1}\xi'_{j\text{-}1}}$), and accounting for the last appended O-labeled tokens, as well as the dependency between entity types ξ' and ξ'' with the O-labeled tokens in-between.

For the last sub-problem, since the length of the O-labeled token sequence exceeds the limit m_O, no dynamic potential would be applied to this O-labeled sequence. This last sub-problem can also be tackled efficiently if we additionally keep the result calculated for the set of all possible labelings up to position $(i - m_O)$, which each token from position $(i - m_O)$ to i will be

labeled as O, i.e. the O-labeled token sequence begins at or before position $(i - m_O)$ and ends at or after position i. Then, by using this calculation result and accounting for the newly appended O-labeled tokens and the last transition from O to ξ'', the last sub-problem can be solved easily. The additionally kept result can be reused to calculate the one of the same kind for the position $(i + 1)$. This is achieved by taking into account the result calculated up to position $(i - m_O)$ for the set of labelings that an O-labeled sequence has begun at position $(i - m_O)$ and will be longer than m_O tokens (so that dynamic potentials need not to be considered for this O-labeled sequence). This calculation is simple and will be shown later in the detailed formulation.

The main principle of our proposed framework has now been explained. The detailed and precise formulations will be derived below.

3.2.2 The Detailed Formulation

The Labeling Scheme

The labeling scheme of our proposed framework is slightly different from that of the original CRF for NER tasks. In our formulation, *all* tokens within an ξ entity are tagged with the I-tags (i.e. ξ_I), while the beginning token and the ending token of the entity are *additionally* tagged with the B-tag (i.e.

ξ_B) and the E-tag (i.e. ξ_E) respectively. So, ξ_B is like an open bracket while ξ_E is like a close bracket, and they together are able to define the boundaries of any entity clearly. In this way, a single-token entity is no longer needed to be tagged by a special S-tag (i.e. ξ_S), but can be tagged by ξ_B and ξ_E together. This labeling scheme is obviously more natural and reasonable for NER tasks, because it labels all entities in a unified manner, regardless of their length. Recall that the original CRF allows only one tag per state, so features for a state tagged with ξ_B or ξ_E are not activated for a state tagged with ξ_S, even though they share many similarities. By using the new labeling, the formulation of our proposed model can also be simplified.

The Modeling

In the formulation of our proposed framework, the notations previously used for the original CRF will be reused here. Also, the following notations are additionally defined:

- $\mathcal{L}$ denotes the set of all labels, including the O label and all entity labels, i.e. {Per, Loc, Org, O}.
- ξ_B, ξ_I and ξ_E denote the B-, I- and E-tags for the entity type ξ respectively.
- m_ξ denotes the maximum length of an entity of type ξ which is allowed to use dynamic potentials ($m_\xi \geq 1$).

- m_O denotes the maximum length of a sequence of O tokens which is allowed to use dynamic potentials ($m_O \geq 1$).

All the notations of potentials for original CRF will be directly adopted here. To recall:

- $\Psi(i, y)$ denotes the potential occurring at position i which depends only on y. This kind of potential represents the weighted score of the state features activated at position i for y. (See Eq. 3.1)
- $\Psi(i, y' \rightarrow y)$ denotes the potential occurring at position i which depends on *both* y' and y. This kind of potential represents the weighted score of the transition features activated at position i for the transition from y' to y. (See Eq. 3.2)
- $\Psi(i, y'y)$ is the composite potential that represents the product of the above two potentials. (See Eq. 3.3)

In addition, some more notations are defined for the dynamic potentials:

- $\Psi(\xi[j, i])$ denotes the dynamic potential that applies to the entity candidate of type ξ which begins at position j and ends at position i, without any restriction on the preceding label. This accounts for the long range dependencies over the whole entity candidate.
- $\Psi(l \rightarrow \xi[j, i])$ denotes the dynamic potential that applies to the entity candidate of type ξ which begins at position j and ends at position

i, with l being the preceding label. This accounts for the long range dependencies between the preceding label l and the whole ξ entity candidate.

- $\Psi(l\xi[j,i])$ denotes the composite dynamic potential that represents the product of the above two potentials.
- $\Psi(\xi' O[j,i]\xi'')$ denotes the dynamic potential linking two entity candidates with a sequence of O-labeled tokens in-between, which begins at position j and ends at position i, and the preceding entity is of type ξ' while the succeeding one is of type ξ''.

The notation d_k^{Ent} denotes the k^{th} feature for the dynamic potentials applied to a whole entity candidate, and $d_{k_1}^{Ent}(\xi[j,i],\boldsymbol{x})$ gives its value for the potential $\Psi(\xi[j,i])$ while $d_{k_2}^{Ent}(l \rightarrow \xi[j,i],\boldsymbol{x})$ gives its value for the potential $\Psi(l \rightarrow \xi[j,i])$. Therefore, we have the following formulae:

$$\Psi(\xi[j,i]) = \exp\left\{\sum_k \lambda_{d_k^{Ent}} d_k^{Ent}(\xi[j,i],\boldsymbol{x})\right\} \text{ , where } i \geq j \geq 1, \tag{3.13}$$

and

$$\Psi(l \rightarrow \xi[j,i]) = \exp\left\{\sum_k \lambda_{d_k^{Ent}} d_k^{Ent}(l \rightarrow \xi[j,i],\boldsymbol{x})\right\} \text{ , where } i \geq j > 1. \tag{3.14}$$

The notation d_k^O denotes the k^{th} feature for the dynamic potentials linking two entity candidates with a sequence of O-labeled tokens in-between, and

$d_k^O(\xi' O[j,i]\xi'', \boldsymbol{x})$ gives its value for the potential $\Psi(\xi' O[j,i]\xi'')$. Therefore, we have the following formula:

$$\Psi(\xi' O[j,i]\xi'') = \exp\left\{\sum_k \lambda_{d_k^O} d_k^O(\xi' O[j,i]\xi'', \boldsymbol{x})\right\} \text{ , where } 1 < j \leq i < |\boldsymbol{x}|. \tag{3.15}$$

To make the notations of the feature functions for dynamic potentials more uniform, we also use the notation $d_k^{Ent}(i, j, \boldsymbol{y}, \boldsymbol{x})$ to represent the original $d_k^{Ent}(\xi[j,i], \boldsymbol{x})$ or $d_k^{Ent}(l \rightarrow \xi[j,i], \boldsymbol{x})$, and use $d_k^O(i, j, \boldsymbol{y}, \boldsymbol{x})$ to represent the original $d_k^O(\xi' O[j,i]\xi'', \boldsymbol{x})$. Then, we use the feature function vector $\boldsymbol{d}(i, j, \boldsymbol{y}, \boldsymbol{x})$ to group all these dynamic potential feature functions.

Recall that a global feature vector $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$ is defined in the formulation of the original CRF to account for all the feature function vectors in the whole sequence, i.e.

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \boldsymbol{f}(i, \boldsymbol{y}, \boldsymbol{x})$$

where $\boldsymbol{f}(i,\ \boldsymbol{y},\ \boldsymbol{x})$ is the vector of feature functions for the usual potentials (i.e. the non-dynamic ones) at position i.

For the formulation of our proposed framework, we should also account for the feature functions for dynamic potentials. Therefore, the global feature vector $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$ in our framework is defined as:

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \begin{pmatrix} \boldsymbol{f}(i, \boldsymbol{y}, \boldsymbol{x}) \\ \sum_j \boldsymbol{d}(i, j, \boldsymbol{y}, \boldsymbol{x}) \end{pmatrix}$$

i.e. the upper part of the global feature vector $\boldsymbol{F}$ is the sum of the original $\boldsymbol{f}$ while the lower part is the sum of $\boldsymbol{d}$.

By redefining $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$, the derivations of the conditional probability distribution $p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x})$, the log-likelihood $\mathcal{Q}_{\boldsymbol{\lambda}}$ and the gradient of $\mathcal{Q}_{\boldsymbol{\lambda}}$ for our proposed framework become the same as those for the original CRF (see Eq. 3.5, 3.8 and their explanation in Section 3.1.2).

The Viterbi Algorithm

For the Viterbi algorithm, the following notations are defined:

- $\delta(i, \xi_E)$, or simply $\delta(i, \xi)$, denotes the maximum score among all possible labelings up to position i where an entity of type ξ ended. In other words, the labeling set under consideration is $\mathrm{X}_{\boxed{i}}^{\boxed{i}}{}_{\xi_i}$.
- $\delta(i, l \rightarrow \xi'')$ denotes the maximum score among all possible labelings up to the transition from position i to $(i{+}1)$, which an l label *ended* at position i and a ξ entity will start at position $(i{+}1)$. When l equals to the O label, the labeling set under consideration is $\mathrm{X}_{\boxed{i}}^{\boxed{i}}{}_{\mathrm{O}_i}\xi''$.
- $\delta_c(i{-}m_\xi, \xi_I)$, or simply $\delta_c(i{-}m_\xi, \xi)$, denotes the maximum score among all possible labelings up to position $(i - m_\xi)$ which the tokens from $(i - m_\xi)$ to i are to be labeled as *one* ξ entity, i.e. the ξ entity begins at or before $(i{-}m_\xi)$ and ends at or after i. This variable refers to the additionally kept result for efficiently resolving the long-range dependencies

linking across an entity, which is explained in Section 3.2.1.

- $\delta_c(i - m_O, O)$ denotes the maximum score among all possible labelings up to position $(i - m_O)$ which the tokens from $(i - m_O)$ to i are all to be labeled as O, i.e. the O-labeled token sequence begins at or before $(i - m_O)$ and ends at or after i. This variable refers to the additionally kept result for efficiently resolving the long-range dependencies linking two entities with a number of O-labeled tokens in-between, which is also explained in Section 3.2.1.

By computing these variables from the beginning to the end of a sentence, the labeling with the maximum score can be obtained. The precise formulae for the computation of the variables are given in the following. (For convenience, we will use $\Psi(j, \xi_{BI})$ as shorthand for "$\Psi(j, \xi_B) \cdot \Psi(j, \xi_I)$".)

$$\delta(i, \xi_E) = \begin{cases} \text{undefined} & \text{, for } i \leq 0 \\ \max \left\{ \begin{array}{l} \max\limits_{k=1..(i-1)} \left\{ \begin{array}{l} \max\limits_{l \in \mathcal{L}} \{\delta(i-k, l \to \xi) \Psi(l\xi[i-k+1, i])\} \\ \quad \cdot \Psi(i-k+1, \xi_{BI}) \prod\limits_{j=i-k+2}^{i} \Psi(j, \xi_I \xi_I) \end{array} \right\}, \\ \Psi(\xi[1,i]) \Psi(1, \xi_B) \Psi(1, \xi_I) \cdot \prod\limits_{j=2}^{i} \Psi(j, \xi_I \xi_I) \end{array} \right\} \cdot \Psi(i, \xi_E) & \text{, for } 1 \leq i \leq m_\xi \\ \max \left\{ \begin{array}{l} \delta_c(i - m_\xi, \xi_I) \prod\limits_{j=i-m_\xi+1}^{i} \Psi(j, \xi_I \xi_I), \\ \max\limits_{k=1..m_\xi} \left\{ \begin{array}{l} \max\limits_{l \in \mathcal{L}} \{\delta(i-k, l \to \xi) \Psi(l\xi[i-k+1, i])\} \\ \quad \cdot \Psi(i-k+1, \xi_{BI}) \prod\limits_{j=i-k+2}^{i} \Psi(j, \xi_I \xi_I) \end{array} \right\} \end{array} \right\} \cdot \Psi(i, \xi_E) & \text{, for } i > m_\xi \end{cases}$$

$$\delta_c(i{-}m_\xi, \xi_I) = \begin{cases} \text{undefined} & \text{, for } i - m_\xi \leq 0 \\ \Psi(1, \xi_{BI}) & \text{, for } i - m_\xi = 1 \\ \max \left\{ \begin{array}{l} \delta_c(i - m_\xi - 1, \xi_I)\Psi(i - m_\xi, \xi_I\xi_I), \\ \max\limits_{l \in \mathcal{L}} \{\delta(i - m_\xi - 1, l \to \xi)\Psi(i - m_\xi, \xi_{BI})\} \end{array} \right\} & \text{, for } i - m_\xi > 1 \end{cases}$$

$$\delta(i, \xi \to \xi'') = \begin{cases} \text{undefined} & \text{, for } i \leq 0 \\ \delta(i, \xi)\Psi(i+1, \xi_E \to \xi''_B) & \text{, for } i \geq 1 \end{cases}$$

$$\delta(i, O \to \xi'') = \begin{cases} \text{undefined} & \text{, for } i \leq 0 \\ \max \left\{ \begin{array}{l} \max\limits_{k=1..(i-1)} \left\{ \begin{array}{l} \max\limits_{\xi' \in \mathcal{L}-\{O\}} \{\delta(i-k, \xi')\Psi(\xi' O[i-k+1, i]\xi'') \\ \quad \cdot \Psi(i-k+1, \xi'_E O)\} \cdot \prod\limits_{j=i-k+2}^{i} \Psi(j, OO) \end{array} \right\}, \\ \Psi(1, O) \cdot \prod\limits_{j=2}^{i} \Psi(j, OO) \end{array} \right\} \cdot \Psi(i+1, O \to \xi''_B) & \text{, for } 1 \leq i \leq m_O \\ \max \left\{ \begin{array}{l} \delta_c(i - m_O, O) \prod\limits_{j=i-m_O+1}^{i} \Psi(j, OO), \\ \max\limits_{k=1..m_O} \left\{ \begin{array}{l} \max\limits_{\xi' \in \mathcal{L}-\{O\}} \{\delta(i-k, \xi')\Psi(\xi' O[i-k+1, i]\xi'') \\ \quad \cdot \Psi(i-k+1, \xi'_E O)\} \cdot \prod\limits_{j=i-k+2}^{i} \Psi(j, OO) \end{array} \right\} \end{array} \right\} \cdot \Psi(i+1, O \to \xi''_B) & \text{, for } i > m_O \end{cases}$$

$$\delta_c(i{-}m_O, O) = \begin{cases} \text{undefined} & \text{, for } i - m_O \leq 0 \\ \Psi(1, O) & \text{, for } i - m_O = 1 \\ \max \left\{ \begin{array}{l} \delta_c(i - m_O - 1, O)\Psi(i - m_O, OO), \\ \max\limits_{\xi' \in \mathcal{L}-\{O\}} \{\delta(i - m_O - 1, \xi')\Psi(i - m_O, \xi' O)\} \end{array} \right\} & \text{, for } i - m_O > 1 \end{cases}$$

Finally, the maximum score among all labelings is $\max\limits_{l \in \mathcal{L}} \{\delta(|\boldsymbol{x}|, l \to \texttt{EoS})\}$

$$\text{, where } \begin{cases} \Psi(|\boldsymbol{x}| + 1, l \to \texttt{EoS}) = 1 & \text{, } \forall l \in \mathcal{L} \\ \Psi(\xi' O[j, i]\texttt{EoS}) = 1 & \text{, } \forall j \in \mathbb{N} \text{ and } \forall \xi' \in \mathcal{L} - \{O\} \end{cases}$$

Note that "EoS" denotes End-of-Sentence and serves as a dummy variable here.

The Forward-Backward Algorithm

The main purpose of the forward-backward algorithm is to calculate the marginal distributions at each position in order to estimate the model parameters. This algorithm is composed of two procedures, i.e. the forward procedure and the backward procedure. In practice, the forward procedure is done first and the values of all the forward variables are cached. Then, in the backward procedure, when the values of the backward variables are being computed at each position, the marginal distributions are computed on the way by utilizing together the cached forward variables and the newly computed backward variables. The following shows the formulations for both procedures.

In principle, the only difference between the forward algorithm (i.e. the algorithm of the forward procedure) and the Viterbi algorithm is that, the forward algorithm computes the accumulative score while the Viterbi algorithm computes the maximum score. Therefore, the formulation of the forward algorithm can be obtained after the following simple modifications are done to the new Viterbi algorithm:

1. Replacing the phrase "the maximum score among all possible labelings up to position ..." by "the accumulative score of all possible labelings up to position ..." in the notation definitions.

2. Replacing δ by α in all notations.

3. Replacing the "max" operation by summation (i.e. $\sum$) in all formulae.

After the above modifications, the last computation result we obtain is no longer that maximum score among all labelings, but the accumulative score of all labelings, i.e. the normalization factor $Z_{\boldsymbol{\lambda}}(\boldsymbol{x})$, which is needed to calculate marginal probabilities in the backward procedure.

The backward algorithm (i.e. the algorithm for the backward procedure) is similar to the forward algorithm, but the job is being done in the reverse order, i.e. the accumulative scores are computed from the end of the sentence rather than from the beginning. For the backward algorithm, the following notations are defined:

- $\beta(i, \xi_E)$, or simply $\beta(i, \xi)$, denotes the accumulative score of all possible labelings down to (but not include) position i where an entity of type ξ ends. In other words, the labeling set under consideration is the "reverse" of the one considered for $\alpha(i, \xi)$. Using the graphical notation, the labeling set is $\boxed{\cdots\, \xi\, i}$X.

- $\beta_c(i - m_\xi, \xi_I)$, or simply $\beta_c(i - m_\xi, \xi)$, denotes the accumulative score among all possible labelings down to (but not include) position $(i - m_\xi)$, in which the tokens from position $(i - m_\xi)$ to position i are labeled as *one* ξ entity, i.e. the ξ entity begins at or before $(i - m_\xi)$ and ends at or

after i. Similarly, the labeling set under consideration is the "reverse" of the one considered for $\alpha_c(i - m_\xi, \xi_I)$.

- $\beta_c(i - m_O, O)$ denotes the accumulative score of all possible labelings down to position $(i - m_O)$, in which the tokens from position $(i - m_O)$ to position i are all labeled as O, i.e. the O-labeled token sequence begins at or before $(i - m_O)$ and ends at or after i. The labeling set under consideration is the "reverse" of the one considered for $\alpha_c(i - m_O, O)$.

- $\beta_*(l\xi[j, i])$ denotes the accumulative score of all possible labelings down to position j, in which a ξ entity starts at position j and ends at position i and follows an l label. Note that the score includes accounting for all the dynamic potentials over the ξ entity.

- $\beta_*(l\xi[j, i] >)$ has a very similar meaning as the previous notation. The only difference is that the ending position of the ξ entity is at *or after* position i.

- $\beta_*(\xi' O[j, i] >)$ denotes the accumulative score of all possible labelings down to position j, in which a sequence of O-labeled tokens starts at position j and ends at or after position i. Note that the score includes accounting for all the dynamic potentials over the sequence of O-labeled tokens.

- $\beta_*(\xi' O[j,i]\xi'')$ has a similar meaning as the previous notation. The only difference is that the sequence of O-labeled tokens ends definitely at position i and is followed by a ξ'' entity.

Note that, in practice, each backward variable marked with an asterisk need not to be stored for more than one position at a time.

The precise formulae for the computation of the above variables are given in the following.

$$\beta(i,\xi_E) = \begin{cases} \text{undefined} & \text{, for } i > |\boldsymbol{x}| \\ 1 & \text{, for } i = |\boldsymbol{x}| \\ \sum\limits_{l\in\mathcal{L}}\{\Psi(i+1,\xi\rightarrow l)\beta_*(\xi l[i+1,i+1]>)\} & \text{, for } 1 \le i < |\boldsymbol{x}| \end{cases}$$

$$\beta_c(i-m_\xi,\xi_I) = \begin{cases} \prod\limits_{j=|\boldsymbol{x}|-m_\xi+1}^{|\boldsymbol{x}|} \Psi(j,\xi_I\xi_I)\Psi(|\boldsymbol{x}|,\xi_E) & \text{, for } i-m_\xi = |\boldsymbol{x}|-m_\xi \\ \left(\begin{array}{l}\Psi(i-m_\xi+1,\xi_I\xi_I)\beta_c(i-m_\xi+1,\xi_I) \\ +\prod\limits_{j=i-m_\xi+1}^{i}\Psi(j,\xi_I\xi_I)\Psi(i,\xi_E)\beta(i,\xi_E)\end{array}\right) & \text{, for } 0 < i-m_\xi < |\boldsymbol{x}|-m_\xi \\ \text{undefined} & \text{, for } i-m_\xi > |\boldsymbol{x}|-m_\xi \end{cases}$$

$$\beta_*(l\xi[j,i]) = \Psi(l\xi[j,i])\Psi(j,\xi_{BI}) \prod_{k=j+1}^{i} \Psi(k,\xi_I\xi_I)\Psi(i,\xi_E)\beta(i,\xi_E) \quad \text{, for } j \ge 1$$

$$\beta_*(l\xi[j,i]>) = \begin{cases} \beta_*(l\xi[j,i]) & \text{, for } i = |\boldsymbol{x}| \\ \beta_*(l\xi[j,i+1]>) + \beta_*(l\xi[j,i]) & \text{, for } j > i-m_\xi+1 \ \&\ 1 \le i < |\boldsymbol{x}| \\ \Psi(j,\xi_{BI})\beta_c(i+1-m_\xi,\xi_I) + \beta_*(l\xi[j,i]) & \text{, for } j = i-m_\xi+1 \ \&\ 1 \le i < |\boldsymbol{x}| \end{cases}$$

Note that when j=1, all l's in the above two formulae vanish.

$$\beta_c(i-m_O, O) = \begin{cases} \prod_{j=|\boldsymbol{x}|-m_O+1}^{|\boldsymbol{x}|} \Psi(j, OO) & \text{, for } i - m_O = |\boldsymbol{x}| - m_O \\ \left(\begin{array}{l} \Psi(i-m_O+1, OO)\beta_c(i-m_O+1, O) + \prod_{j=i-m_O+1}^{i} \Psi(j, OO) \\ \quad \cdot \sum_{\xi \in \mathcal{L}-\{O\}} \{\Psi(i+1, O \to \xi)\beta_*(O\xi[i+1, i+1] >)\} \end{array} \right) & \text{, for } 0 < i - m_O < |\boldsymbol{x}| - m_O \\ \text{undefined} & \text{, for } i - m_O > |\boldsymbol{x}| - m_O \end{cases}$$

$$\beta_*(\xi' O[j, i]\xi'') = \begin{cases} \left(\begin{array}{l} \Psi(\xi' O[j, i]\xi'')\Psi(j, O) \prod_{k=j+1}^{i} \Psi(k, OO) \\ \quad \cdot \Psi(i+1, O \to \xi'')\beta_*(O\xi''[i+1, i+1] >) \end{array} \right) & \text{, for } j > 1 \\ \Psi(j, O) \prod_{k=j+1}^{i} \Psi(k, OO)\Psi(i+1, O \to \xi'')\beta_*(O\xi''[i+1, i+1] >) & \text{, for } j = 1 \end{cases}$$

$$\beta_*(\xi' O[j, i] >) = \begin{cases} \Psi(j, O) \prod_{k=j+1}^{|\boldsymbol{x}|} \Psi(k, OO) & \text{, for } i = |\boldsymbol{x}| \\ \beta_*(\xi' O[j, i+1] >) + \sum_{\xi'' \in \mathcal{L}-\{O\}} \beta_*(\xi' O[j, i]\xi'') & \text{, for } j > i - m_O + 1 \;\&\; 1 \leq i < |\boldsymbol{x}| \\ \Psi(j, O)\beta_c(i+1-m_O, O) + \sum_{\xi'' \in \mathcal{L}-\{O\}} \beta_*(\xi' O[j, i]\xi'') & \text{, for } j = i - m_O + 1 \;\&\; 1 \leq i < |\boldsymbol{x}| \end{cases}$$

Note that when j=1, all ξ' in the above two formulae vanish.

Computing Marginal Distributions

The main purpose of the forward-backward algorithm is to facilitate the calculation of various kinds of marginal probabilities, such that the expected values of the feature functions under the model distribution can be obtained. In this way, the gradient of the log-likelihood can be known and the log-likelihood of the training data can be maximized.

There are several kinds of feature functions and they need different kinds of marginal probabilities. The following lists out each kind of the feature functions and the calculation required to obtain the corresponding marginal

probabilities. To keep the formulae simple, we just calculate the marginal likelihood scores and the actual marginal probability can be obtained by dividing the scores by the normalization factor $Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k)$.

- Transition feature $t(i, l \rightarrow \xi_B, \boldsymbol{x})$, for $2 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(l \text{ ends at } i\text{-1 and } \xi_B \text{ at } i) = \alpha(i-1, l \rightarrow \xi_B) \cdot \beta_*(l\xi[i,i] >) \tag{3.16}$$

- Transition feature $t(i, \xi_I \rightarrow \xi_I, \boldsymbol{x})$, for $2 \leq i \leq |\boldsymbol{x}|$, requires:

$$\begin{aligned} &\text{score}(\text{tokens at } i\text{-1 and } i \text{ are in } \textit{one } \xi \text{ entity}) \\ &= \begin{cases} \beta_*(\xi[1,i] >) + \sum_{j=2}^{i-1} \sum_{l \in \mathcal{L}} \{\alpha(j-1, l \rightarrow \xi) \beta_*(l\xi[j,i] >)\} & , \text{for } 2 \leq i \leq m_\xi \\ \left(\begin{array}{l} \alpha_c(i - m_\xi, \xi_I) \beta_c(i - m_\xi, \xi_I) \\ \quad + \sum_{j=i-m_\xi+1}^{i-1} \sum_{l \in \mathcal{L}} \{\alpha(j-1, l \rightarrow \xi) \beta_*(l\xi[j,i] >)\} \end{array} \right) & , \text{for } m_\xi < i \leq |\boldsymbol{x}| \end{cases} \end{aligned} \tag{3.17}$$

- State feature $s(i, \xi_B, \boldsymbol{x})$, for $1 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(\xi_B \text{ at } i) = \begin{cases} \beta_*(l\xi[i,i] >) & , \text{for } i = 1 \\ \sum_{l \in \mathcal{L}} \text{score}(l \text{ ends at } i\text{-1 and } \xi_B \text{ at } i) & , \text{for } i > 1 \end{cases} \tag{3.18}$$

- State feature $s(i, \xi_I, \boldsymbol{x})$, for $1 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(\xi_I \text{ at } i) = \begin{cases} \beta_*(\xi[i,i] >) & , \text{for } i = 1 \\ \left(\begin{array}{r} \text{score}(\text{tokens at } i\text{-1 and } i \text{ are in } \textit{one } \xi \text{ entity}) \\ + \sum_{l \in \mathcal{L}} \{\alpha(i-1, l \rightarrow \xi) \beta_*(l\xi[i,i] >)\} \end{array} \right) & , \text{for } i > 1 \end{cases} \tag{3.19}$$

- State feature $s(i, \xi_E, \boldsymbol{x})$, for $1 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(\xi_E \text{ at } i) = \alpha(i, \xi_E)\beta(i, \xi_E) \tag{3.20}$$

- Dynamic potential feature $d^{Ent}(l \rightarrow \xi[j, i], \boldsymbol{x})$, for

  $\max(2, i - m_\xi + 1) \leq j \leq i$ and $1 < i \leq |\boldsymbol{x}|$, requires:

$$\begin{aligned} &\text{score}(\text{a } \xi \text{ entity appear from } j \text{ to } i \text{ following a } l \text{ label}) \\ &= \alpha(j-1, l \rightarrow \xi)\beta_*(i, l\xi[j, i]) \end{aligned} \tag{3.21}$$

- Dynamic potential feature $d^{Ent}(\xi[j, i], \boldsymbol{x})$, for

  $\max(1, i - m_\xi + 1) \leq j \leq i$ and $1 \leq i \leq |\boldsymbol{x}|$, requires:

$$\begin{aligned} &\text{score}(\text{a } \xi \text{ entity appear from } j \text{ to } i) \\ &= \begin{cases} \beta_*(\xi[1, i]) & , \text{for } j = 1 \\ \sum_{l \in \mathcal{L}} \text{score}(\text{a } \xi \text{ entity appear from } j \text{ to } i \text{ following a } l \text{ label}) & , \text{for } j > 1 \end{cases} \end{aligned} \tag{3.22}$$

- Transition feature $t(i, \xi'_E \rightarrow O, \boldsymbol{x})$, for $2 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(\xi'_E \text{ at } i\text{-1 and } O \text{ at } i) = \alpha(i-1, \xi'_E)\Psi(i, \xi' \rightarrow O)\beta_*(\xi' O[i, i] >) \tag{3.23}$$

- Transition feature $t(i, O \rightarrow O, \boldsymbol{x})$, for $2 \leq i \leq |\boldsymbol{x}|$, requires:

  score(O at i-1 and O at i)

$$= \begin{cases} \beta_*(O[1,i]>) + \sum_{j=2}^{i-1} \sum_{\xi' \in \mathcal{L}-\{O\}} \{\alpha(j-1, \xi'_E)\Psi(j, \xi' \rightarrow O)\beta_*(\xi' O[j,i]>)\} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{, for } 2 \leq i \leq m_O \\ \left(\begin{array}{l} \alpha_c(i-m_O, O)\beta_c(i-m_O, O) \\ \quad + \sum_{j=i-m_O+1}^{i-1} \sum_{\xi' \in \mathcal{L}-\{O\}} \{\alpha(j-1, \xi'_E)\Psi(\xi' \rightarrow O)\beta_*(\xi' O[j,i]>)\} \end{array} \right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{, for } m_O < i \leq |\boldsymbol{x}| \end{cases} \tag{3.24}$$

- State feature $s(i, O, \boldsymbol{x})$, for $1 \leq i \leq |\boldsymbol{x}|$, requires:

$$\text{score}(O \text{ at } i) = \begin{cases} \beta_*(O[i,i]>) & \text{, for } i = 1 \\ \left(\begin{array}{l} \text{score}(O \text{ at } i\text{-1 and } O \text{ at } i) \\ \quad + \sum_{\xi' \in \mathcal{L}-\{O\}} \text{score}(\xi'_E \text{ at } i\text{-1 and } O \text{ at } i) \end{array} \right) & \text{, for } i > 1 \end{cases} \tag{3.25}$$

- Dynamic potential feature $d^O(\xi' O[j,i]\xi'', \boldsymbol{x})$, for

  $\max(2, i - m_O + 1) \leq j \leq i$ and $1 < i < |\boldsymbol{x}|$, requires:

  score(an O-labeled sequence appears from j to i following ξ' and followed by ξ'')

  $= \alpha(j-1, \xi'_E)\Psi(j, \xi' \rightarrow O)\beta_*(i, \xi' O[j,i]\xi'')$

(3.26)

3.2.3 Adapting Features from Original CRF to CRFDP

Because of the differences in the labeling schemes, we should adapt the features from the original CRF to CRFDP in order to capture almost the same

information as in the original CRF.

This arrangement is mainly due to the change of the role of the entity I-tag in our proposed framework, as well as the removal of the entity S-tag and a number of transition types within an entity.

In our proposed framework, the entity I-tag is not only to tag the tokens between the beginning and ending tokens of an entity as in the original CRF, but it is also responsible to tag the beginning and ending tokens. Therefore, unlike the original CRF, the features activated for an entity I-tag, which may be under some conditions on the observations, must also be activated for the states tagged with the corresponding entity B-tag or E-tag under the same conditions in CRFDP. Consider the features for an entity I-tag which uses the information of c_{i-2} or c_{i+2}, i.e. the characters at two positions before or after the current position. Regardless of which frameworks we are considering for, risk exists when using these features because the tags of these characters are not taken into account by these features. In other words, whether these characters are labeled as O, or part of the current entity candidate, or even part of some other entities are not to be considered when they are being used. This can be a potential source of noise to the inference. Because of the change of the role of the entity I-tag, this problem becomes more serious in CRFDP than in the original CRF. In order to eliminate this adverse effect, we restrict the features for entity I-tags in CRFDP from using the information of the

furthest characters used by the features of the same kind in the original CRF.

The removal of the entity S-tag from the formulation of CRFDP potentially causes some information lost, since single-character entities may have some specific probability distributions in a number of aspects. In our original CRF models, mainly the context and the character representing the entity are captured for this kind of entities. Therefore, as a complement to CRFDP, some feature types that utilize short-range dynamic potentials should be applied to capture the similar information for single-character entities.

Also, it should be noted that our proposed framework has fewer types of transitions. Firstly, the removal of the S-tag brings the loss of transitions specifically to and from single-character entities. Secondly, there are no longer B→E, B→I and I→E transitions within an entity. The significance of these transitions is that they implicitly help the original CRF to learn the probability distribution about the length of different kinds of entities. To complement for the information loss in this aspect, we again apply some feature types that utilize short-range dynamic potentials to capture the information about the entity length up to three characters.

In the next chapter, we will present some concrete examples to show how the above principles can be implemented in order to adapt the features.

Chapter 4

Experiments

In this chapter, we will evaluate our proposed model by conducting extensive experiments on a few datasets. For each dataset, we try to compare the performance among the following 3 models: (1) the original CRF with basic features, (2) our proposed framework with *basic* features, and (3) our proposed framework with the *full* incorporation of dynamic potentials. We use "OrigCRF", "CRFDP-basic", and "CRFDP-full" as the abbreviations for these three models respectively. We compare OrigCRF and CRFDP-basic to show that the proposed framework is able to preserve the powerful inference of the original CRF. On the other hand, comparing CRFDP-full against CRFDP-basic can show that dynamic potentials help improve the inference. In the experiments, we have not used any external resources so that the performance of the probabilistic frameworks can be compared fairly.

In addition, for the completeness of this research work, the comparison between CRFDP-full and OrigCRF is included. However, it should be noted that such comparison may not be suitable to show the effectiveness of dynamic potentials, because these two models differ not merely on the utilization of dynamic potentials, but in quite a number of aspects, especially the labeling scheme and software implementation. These differences can affect the performance quite significantly. In contrast, CRFDP-full and CRFDP-basic differ only on the utilization of long-range dynamic potentials. Therefore, the experiments on CRFDP-basic can serve as the control experiments of those on CRFDP-full for this objective.

4.1 Datasets

The first dataset we used is People's Daily (January 1998) corpus, and we use "PDJ98" as its abbreviation in this thesis. This corpus contains the whole month of People's Daily newspapers of the January in 1998, and its content includes all types of articles appeared in the newspaper. The original corpus is designed for general natural language tasks. Therefore, we have done some adjustments on annotation to this corpus for our experiments. Firstly, as the corpus originally annotates all abbreviations using the same label, we distinguished the abbreviated named entities from other abbreviations

and labeled them using the appropriate entity labels. Secondly, since the last name and given name of a Chinese person are separately tagged, we have concatenated them together so as to be consistent with the definition of person names. Thirdly, we removed all information other than named entity labels, mainly word segmentation and part-of-speech tags, since the task is to recognize named entities from plain text sentences.

As both CRF and our proposed framework are sentence-based models, the text documents in the corpus are converted from paragraph-based to sentence-based. The sentence boundaries in a paragraph are simply detected by the full stop punctuation in Chinese, i.e. " ∘ ".

The whole modified corpus has 44010 sentences in total. In order to support the statistical significance about the improvement of our proposed model above the original CRF, we have conducted a 5-fold cross-validation on this corpus. For each fold, one fifth of the corpus, i.e. 8802 sentences which are contiguous in the corpus, are used as the testing set.

The second and the third datasets we used are the MSRA and CityU corpora from the NER task of SIGHAN Bakeoff 2007. Both datasets contain the standard training and testing sets. The text documents in these corpora are already sentence-based. Since the training and testing sets are standard ones, we have kept them and have not further conducted cross-validation.

The statistics summary of the corpora is given in Table 4.1.

				Persons		Locations		Organizations	
		#sent.	#tok.	#ent.	#tok.	#ent.	#tok.	#ent.	#tok.
PDJ98-1	train	35208	1480k	16803	48604	20803	49471	9492	52536
	test	8802	362k	3183	9073	4858	12093	2098	11714
PDJ98-2	train	35208	1469k	16630	48136	20819	50070	9266	50692
	test	8802	373k	3356	9541	4842	11494	2324	13558
PDJ98-3	train	35208	1475k	16215	46721	20666	49397	9342	51985
	test	8802	367k	3771	10956	4995	12167	2248	12265
PDJ98-4	train	35208	1463k	13892	39771	19857	48144	9026	50395
	test	8802	378k	6094	17906	5804	13420	2564	13855
PDJ98-5	train	35208	1480k	16404	47476	20499	49174	9234	51392
	test	8802	362k	3582	10201	5162	12390	2356	12858
MSRA	train	23182	1089k	9028	26623	18522	43634	10261	51895
	test	4636	219k	1864	5465	3658	8606	2185	10941
CityU	train	36334	1772k	16552	49294	36213	82208	13490	35315
	test	8092	382k	4940	14463	4847	11049	3227	8251

(Note: "#sent." is the number of sentences in the dataset, while "#ent." and "#tok." are the number of entities and the number of tokens which belong to the corresponding entity type in the dataset.)

Table 4.1: Statistics summary of the datasets for the experiments

4.2 Features

The set of features used by the original CRF models follows the one described in [8], as it is one of the best models and has achieved stable performance across the datasets in the Third International Chinese Language Processing Bakeoff [16]. The tagging scheme we used in this model is BIOES. The feature types used in the models are listed in Table 4.2. Note that we have not followed [8] to include the feature type (y_i, c_{i-1}, c_{i+1}), since we found that

this feature type may slightly degrade the performance.

Feature type (and description)	Dependent variables for each sub-type	Further conditions for a feature to activate
State unigram joining with local characters	– Y_i, C_{i-2} – Y_i, C_{i-1} – Y_i, C_i – Y_i, C_{i+1} – Y_i, C_{i+2} – Y_i, C_{i-2}, C_{i-1} – Y_i, C_{i-1}, C_i – Y_i, C_i, C_{i+1} – Y_i, C_{i+1}, C_{i+2}	No extra conditions. (Remarks: It means that Y_i can be any possible state)
State bigram	– Y_{i-1}, Y_i	For any possible transition $Y_{i-1} \rightarrow Y_i$

Note 1: Y_i is the random variable (RV) of the state at position i, while C_i is the RV of the Chinese character at position i.

Note 2: Each feature of the above feature types activates only for a particaular tag for each of its Y variables and a particular value for each of its other dependent variables.

Note 3: A feature is not valid if any index of its dependent variables is not in the possible range, i.e. either less than 1 or larger than the length of the current sentence.

Table 4.2: List of feature types employed in OrigCRF

We used CRF++ [14] to conduct experiments on OrigCRF. For each of the datasets, a number of feature cutoff values are investigated and the one that achieves the best F-measure is applied for comparison. This value is 2 for MSRA, while it is 4 for CityU. This value is not the same for each fold of PDJ98. The detailed performance figures of applying different feature cutoff values on these datasets are given in Appendix A. For our proposed

framework, we have not obtained such optimal feature cutoff values. Rather, we only used one single value for each dataset. For MSRA and CityU, the above values are applied directly to our proposed framework, which are not guaranteed to be optimal for the framework. For all the folds of PDJ98, we used the middle value among those tested for the original CRF models, which is 3. As a result, the cutoff values for OrigCRF are favorably optimized for comparison, whereas the cutoff values for our proposed framework may not be optimized. This ensures that the improvement of our proposed framework over the original CRF does not come from the unfavorable setting of the cutoff values for the original CRF models.

We have built two models using our proposed framework, i.e. CRFDP-basic and CRFDP-full as mentioned previously. CRFDP-basic only uses basic features, similar to OrigCRF. Because of the differences in the labeling schemes, we used a slightly different set of features to capture almost the same information as in OrigCRF. All these features are listed in Table 4.3. The reasons for such arrangement are already discussed in Section 3.2.3.

Note that *unsupported features* are used for these feature types for fair comparison, as the software package we used to do experiments for OrigCRF (i.e. CRF++) uses unsupported features. To briefly explain, unsupported features are those activated for wrong labelings but not for the correct labeling, and they are shown to help improve the performance [28].

Feature type (and description)	Dependent variables for each (sub-)type	Further conditions for a feature to activate
Usual state unigram joining with local characters (Unlike OrigCRF, this type only applies to the states tagged with O, B-tag or E-tag in this model.)	– Y_i, C_{i-2} – Y_i, C_{i-1} – Y_i, C_i – Y_i, C_{i+1} – Y_i, C_{i+2} – Y_i, C_{i-2}, C_{i-1} – Y_i, C_{i-1}, C_i – Y_i, C_i, C_{i+1} – Y_i, C_{i+1}, C_{i+2}	Y_i must be tagged with an O label, an entity B-tag or E-tag.
State bigram	– Y_{i-1}, Y_i	For any possible transition $Y_{i-1} \rightarrow Y_i$, where $i > 1$
Entity I-tag unigram & bigram joining with local characters (This type is for adapting the change of the role of entity I-tag)	Unigram: – Y_i, C_i – Y_i, C_{i-1}, C_i – Y_i, C_i, C_{i+1} Bigram: – Y_{i-1}, Y_i, C_{i-1} – Y_{i-1}, Y_i, C_i – Y_{i-1}, Y_i, C_{i-2}, C_{i-1} – Y_{i-1}, Y_i, C_{i-1}, C_i – Y_{i-1}, Y_i, C_i, C_{i+1}	For unigram: Y_i must be tagged with entity I-tag. For bigram: Both Y_{i-1} & Y_i must be tagged with entity I-tag and no entity boundary in-between, where $i > 1$.
Single-character entity joining with local characters (This type is to complement for the removal of entity S-tag.)	– Y_i, C_{i-2} – Y_i, C_{i-1} – Y_i, C_i – Y_i, C_{i+1} – Y_i, C_{i+2} – Y_i, C_{i-2}, C_{i-1} – Y_i, C_{i-1}, C_i – Y_i, C_i, C_{i+1} – Y_i, C_{i+1}, C_{i+2}	The state Y_i itself must form a candidate of single-character entity.
Short entity length, not longer than *3* char. (This is to complement for the loss of some transition types.)	Y_j to Y_i and l, where l is the entity length (i.e. $l = i - j + 1$), with $1 \leq l \leq 3$ and $j \geq 1$	The states Y_j to Y_i must form an entity candidate.

Note 1: Y_i is the random variable (RV) of the state at position i, while C_i is the RV of the Chinese character at position i.

Note 2: Each feature of the first three feature types activates only for a particular tag for each of its Y variables and a particular value for each of its other dependent variables. And each feature of the other feature types activates only for a particular set of values for all of its dependent variables.

Note 3: A feature is not valid if any index of its dependent variables is not in the possible range, i.e. either less than 1 or larger than the length of the current sentence.

Table 4.3: List of basic feature types employed in CRFDP-basic and CRFDP-full

CRFDP-full, on the other hand, uses not only the basic features listed in Table 4.3, except the feature type "short entity length" as it would be replaced by another version, but also some advanced feature types that extensively utilize dynamic potentials. The length limits of the dynamic potentials for concatenating O labels, persons, locations and organizations (i.e. m_{O}, m_{Per}, m_{Loc}, and m_{Org}) are set to be 4, 7, 7, and 15 respectively. These features are listed in Table 4.4.

Note that we did not manually forge the structures for the last feature type "entity structure". Rather, we write a script that scans through the entities in the training data to automatically generate a lot of patterns, and then exhaustively match them with the entities again. Those structures with the number of appearances higher than some threshold values are accepted and used as features. Therefore, no external resource nor linguistic knowledge was used in this feature type so that fair comparisons among the models can be made.

4.3 Evaluation Metrics

In the experiments, we report the results by utilizing the evaluation script provided by SIGHAN Bakeoff 2006 (and also 2007) for the NER task. The script mainly reports 3 types of metrics: precision, recall and F1-score. F1-

Feature type (and description)	Dependent variables for each (sub-)type	Further conditions for a feature to activate
O-labeled character sequence connecting entities	Y_{j-1} to Y_{i+1}, C_j to C_i, and l, where $l = i - j + 1$ and $1 \leq l \leq m_{\mathrm{O}}$	The states from Y_j to Y_i must form an O-labeled character sequence, while both Y_{j-1} and Y_{i+1} must be entity states.
Entity identity	Y_j to Y_i, C_j to C_i, and l, where $l = i - j + 1$ and $1 \leq l \leq m_\xi$	The states from Y_j to Y_i must form an whole entity candidate of type ξ.
Entity length	Y_j to Y_i, and l, where $l = i - j + 1$ and $1 \leq l \leq m_\xi$	The states from Y_j to Y_i must form an whole entity candidate of type ξ.
Entity length joining with the characters in the entity at some specific positions	Y_j to Y_i, l, and the following variables for each sub-type, where $l = i - j + 1$ and $1 \leq l \leq m_\xi$: - C_j - C_{j+1} - C_{j+2} - C_{j+3} - C_i - C_{i-1} - C_{i-2} - C_{i-3} - C_j, C_{j+1} - C_{j+1}, C_{j+2} - C_{j+2}, C_{j+3} - C_{i-1}, C_i - C_{i-2}, C_{i-1} - C_{i-2} to C_i - C_{i-3}, C_{i-2} - C_{i-3} to C_{i-1} - C_{i-3} to C_i	The states from Y_j to Y_i must form an whole entity candidate of type ξ.
Entity structure	Y_j to Y_i, l and S, where $l = i - j + 1$, $1 \leq l \leq m_\xi$, and S represents some structure.	The states from Y_j to Y_i must form an whole entity candidate of type ξ, while the character sequence C_j to C_i must match the structure S.

Note 1: Y_i is the random variable (RV) of the state at position i, while C_i is the RV of the Chinese character at position i.

Note 2: Each feature activates only for a particular set of values for all of its dependent variables.

Note 3: A feature is not valid if any index of its dependent variables is not in the possible range, i.e. either less than 1 or larger than the length of the current sentence.

Table 4.4: List of advanced feature types employed in CRFDP-full

score is the weighted harmonic mean of precision and recall. These metrics are defined as follows:

$$\text{precision} = \frac{\text{no. of correctly recognized NEs}}{\text{no. of recognized NEs}} \tag{4.1}$$

$$\text{recall} = \frac{\text{no. of correctly recognized NEs}}{\text{total no. of true NEs}} \tag{4.2}$$

$$\text{F1-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{4.3}$$

These scores are reported specifically for each type of NEs. The overall scores are reported as well.

4.4 Results and Discussion

We evaluated the performance of OrigCRF, CRFDP-basic and CRFDP-full on each corpus using the mentioned metrics. The detailed tabulation of the results on the corpora MSRA, CityU and PDJ98 are shown in Table 4.5, Table 4.6 and Table 4.7 respectively. Note that the results on PDJ98 are the average over its five folds.

From the results, it can be seen that the OrigCRF and CRFDP-basic have similar performance. It means that, our proposed framework using only the basic features, which only captures the dependencies among local states, can perform as well as the original CRF. This shows that the proposed framework is able to preserve the powerful inference of the original CRF.

	precision	recall	F1-score
OrigCRF			
- location	**93.85%**	88.41%	91.05
- organization	87.95%	80.14%	83.86
- person	**96.65%**	88.20%	92.23
- overall	92.87%	86.01%	89.31
CRFDP-basic			
- location	93.10%	88.96%	90.98
- organization	88.24%	80.00%	83.92
- person	96.63%	89.27%	92.81
- overall	92.61%	86.49%	89.45
CRFDP-full			
- location	93.37%	**90.13%**	**91.72**
- organization	**89.29%**	**80.50%**	**84.67**
- person	96.62%	**90.61%**	**93.52**
- overall	**93.05%**	**87.52%**	**90.20**

Table 4.5: Performance on MSRA corpus

It is also obvious that CRFDP-full generally outperforms CRFDP-basic. CRFDP-full shows consistent improvement on the F1-scores of all entity types (and therefore the overall F1-score) across the corpora. This shows that with the help of dynamic potentials, our proposed framework is able to capture more dependencies and the inference can therefore be enhanced.

We also calculated the statistical significance of such improvement on the PDJ98 corpus. Since PDJ98 does not come with official training set and testing set, we have done a paired t-test on its five folds. We found that, at the significance level of 0.5%, it can be concluded that CRFDP-full has improvement over CRFDP-basic for PDJ98 on the overall F1-score and recall, as well as the F1-scores and recall values of both location and person entity

	precision	recall	F1-score
OrigCRF			
- location	**88.45%**	84.96%	86.67
- organization	87.46%	54.48%	67.14
- person	92.51%	73.04%	81.63
- overall	89.76%	72.88%	80.44
CRFDP-basic			
- location	88.32%	85.00%	86.63
- organization	**88.03%**	54.45%	67.28
- person	**92.72%**	74.25%	82.46
- overall	**89.90%**	73.34%	80.78
CRFDP-full			
- location	87.72%	**86.22%**	**86.96**
- organization	86.61%	**55.31%**	**67.51**
- person	92.42%	**76.26%**	**83.56**
- overall	89.27%	**74.77%**	**81.38**

Table 4.6: Performance on CityU corpus

types. The detailed calculation of the paired t-tests is shown in Appendix B.

One may notice that the recall values of CRFDP-full are generally higher than those of CRFDP-basic while no similar boost has been given to the precision values. Such phenomenon is actually due to the types of dynamic potential features that CRFDP-full used, i.e. those listed in Table 4.4. Note that for most of these features, the activation conditions are not restrictive, and therefore tends to raise the recall but not the precision. Precision can be raised if a set of more restrictive features is used instead.

For the completeness of this research work, we also compare the performance of CRFDP-full and OrigCRF here. Note that such comparison may not be suitable to show the effectiveness of dynamic potentials, as explained

	precision	recall	F1-score
OrigCRF			
- location	**91.11%**	83.85%	87.33
- organization	**89.90%**	**82.19%**	**85.86**
- person	**95.32%**	83.16%	88.82
- overall	**92.28%**	83.32%	87.57
CRFDP-basic			
- location	89.94%	84.15%	86.95
- organization	89.20%	80.89%	84.84
- person	94.70%	84.13%	89.09
- overall	91.40%	83.52%	87.28
CRFDP-full			
- location	90.59%	**85.31%**	**87.87**
- organization	89.88%	81.66%	85.56
- person	94.83%	**85.76%**	**90.05**
- overall	91.88%	**84.77%**	**88.18**

Table 4.7: Performance on PDJ98 corpus

in the beginning of this chapter. However, as expected, the result of this comparison is very similar to that of CRFDP-full and CRFDP-basic.

Generally, CRFDP-full outperforms OrigCRF. It consistently has higher overall F1-scores across the corpora. For the nine F1-scores of the entity types for the three corpora, CRFDP-full has eight of them higher than OrigCRF. We have also done a paired t-test on PDJ98, and found that at the significance level of 0.5%, it can be concluded that CRFDP-full has improvement over OrigCRF for PDJ98 on the overall F1-score and recall, as well as the F1-scores and recall values of both location and person entity types. The detailed calculation of this paired t-tests is shown in Appendix C.

Chapter 5

Conclusions and Future Work

In recent years, the performance of Chinese NER has been improved significantly by utilizing various probabilistic frameworks. However, the development of Chinese NER still has room for improvement especially when comparing with the performance of English NER. The reason is that most of the probabilistic frameworks fail to capture some specific characteristics of Chinese language effectively, as they are developed mostly for western languages. The linear-chain CRF, which has reported the best performance on Chinese NER, also suffer from this problem. We address this issue by improving the modeling of the linear-chain CRF.

In this research work, we have extended the linear-chain CRF by introducing dynamic potentials, which enable the framework to capture the dependencies across a number of states. To keep the inference efficient, we

have also adapted the common Viterbi and forward-backward algorithms from the original CRF. Our experimental result shows that the newly formulated framework, i.e. CRFDP, has improvement over the original CRF. Experimental result shows that such improvement is of high statistical significance and consistent over several datasets. In our detailed investigation, we have found that without using long-range dynamic potentials, CRFDP performs similarly with the original CRF. This shows that CRFDP is able to preserve the powerful inference of the linear-chain CRF. Using a set of control experiments, we have also verified that dynamic potentials are effective in improving the inference.

This research work demonstrates how to apply dynamic potentials in CRFs for Chinese NER tasks. In fact, many natural language processing tasks can benefit from the concept of dynamic potential, especially the Chinese language tasks. A possible direction of future research is to apply dynamic potentials for the general labeling problem. This requires generalizing the principle of dynamic potentials in linear-chain CRFs, as well as formulating the general algorithms that do not subject to any specific kind of long-range state dependencies.

Bibliography

[1] G. Andrew. A Hybrid Markov/Semi-Markov Conditional Random Field for Sequence Segmentation. *Proceedings of EMNLP*, pages 465–472, 2006.

[2] A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] D.M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201, 1997.

[4] D.M. Bikel, R. Schwartz, and R.M. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning*, 34(1):211–231, 1999.

[5] A. Borthwick. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, New York University, 1999.

[6] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 152–160, 1998.

[7] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. NYU: Description of the MENE Named Entity System as Used in MUC-7. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 6, 1998.

[8] Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 173–176, Sydney, Australia, July 2006. Association for Computational Linguistics.

[9] Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. Chinese named entity recognition with conditional random fields. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 118–121, Sydney, Australia, July 2006. Association for Computational Linguistics.

[10] H.L. Chieu and H.T. Ng. Named entity recognition with a maximum entropy approach. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 160–163, 2003.

[11] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171, 2003.

[12] G. Fu and K.K. Luke. Chinese named entity recognition using lexicalized HMMs. *ACM SIGKDD Explorations Newsletter*, 7(1):19–25, 2005.

[13] H. Jing, R. Florian, X. Luo, T. Zhang, and A. Ittycheriah. Howtoge-

taChineseName (Entity): Segmentation and Combination Issues. *Proceedings of EMNLP'03*, pages 200–207, 2003.

[14] T. Kudo. CRF++: Yet Another CRF toolkit, 2005.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[16] G.A. Levow. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, 2006.

[17] Lishuang Li, Tingting Mao, Degen Huang, and Yuansheng Yang. Hybrid models for chinese named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 72–78, Sydney, Australia, July 2006. Association for Computational Linguistics.

[18] P. Liang. *Semi-Supervised Learning for Natural Language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[19] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191, 2003.

[20] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weischedel. Algorithms that learn to extract information-BBN: Description of the SIFT system as used for MUC-7. *Proceedings of MUC*, 7, 1998.

[21] T. Minka. Discriminative models, not discriminative training. Technical report, Technical Report MSR-TR-2005-144, Microsoft Research, October 2005. ftp://ftp. research. microsoft. com/pub/tr/TR-2005-144. pdf.

[22] A. Ng and M. Jordan. On generative vs. discriminative classifiers: A comparison of logistic regression and naive bayes. *Proc. of Advances in Neural Information Processing*, 15, 2002.

[23] F. Peng, F. Feng, and A. McCallum. Chinese segmentation and new word detection using conditional random fields. *COLING 2004*, pages 562–568, 2004.

[24] A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996.

[25] E.F.T.K. Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147, 2003.

[26] S. Sarawagi. Efficient inference on sequence segmentation models. *Proceedings of the 23rd international conference on Machine learning*, pages 793–800, 2006.

[27] S. Sarawagi and W.W. Cohen. Semi-markov conditional random fields for information extraction. *Advances in Neural Information Processing Systems*, 17:1185–1192, 2005.

[28] F. Sha and F. Pereira. Shallow parsing with conditional random fields. *Proceedings of the 2003 Conference of the North American Chapter*

of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 134–141, 2003.

[29] J. Sun, J. Gao, L. Zhang, M. Zhou, and C. Huang. Chinese named entity identification using class-based language model. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7, 2002.

[30] C. Sutton and A. McCallum. Collective segmentation and labeling of distant entities in information extraction. *ICML workshop on Statistical Relational Learning*, 2004.

[31] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[32] Chia-Wei Wu, Shyh-Yi Jan, Richard Tzong-Han Tsai, and Wen-Lian Hsu. On using ensemble methods for chinese named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 142–145, Sydney, Australia, July 2006. Association for Computational Linguistics.

[33] Y. Wu, J. Zhao, and B. Xu. Chinese Named Entity Recognition Model Based on Multiple Features. *the Proceeding of HLT/EMNLP*, pages 427–434, 2005.

[34] Yu-Chieh Wu, Jie-Chi Yang, and Qian-Xiang Lin. Description of the ncu chinese word segmentation and named entity recognition system for sighan bakeoff 2006. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 209–212, Sydney, Australia, July 2006. Association for Computational Linguistics.

[35] X. Yu, W. Lam, S. Chan, Y.K. Wu, and B. Chen. Chinese NER Using CRFs and Logic for the Fourth SIGHAN Bakeoff. *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, pages 102–105, 2008.

[36] Xiaofeng Yu, Marine Carpuat, and Dekai Wu. Boosting for chinese named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 150–153, Sydney, Australia, July 2006. Association for Computational Linguistics.

[37] H.P. Zhang, Q. Liu, H. Yu, X. Cheng, and S. Bai. Chinese Named Entity Recognition Using Role Model. *Computational Linguistics and Chinese Language Processing*, 8(2):29–60, 2003.

[38] Suxiang Zhang, Ying Qin, Juan Wen, and Xiaojie Wang. Word segmentation and named entity recognition for sighan bakeoff3. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 158–161, Sydney, Australia, July 2006. Association for Computational Linguistics.

[39] Y.F.L.S.J. Zhang. Early Results for Chinese Named Entity Recognition Using Conditional Random Fields Model, HMM and Maximum Entropy. *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 549–552, 2005.

[40] Junsheng Zhou, Liang He, Xinyu Dai, and Jiajun Chen. Chinese named entity recognition with a multi-phase model. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 213–216,

Sydney, Australia, July 2006. Association for Computational Linguistics.

Appendix A

This appendix depicts the tables showing the performance of the original CRF models when applying different feature cutoff values c on each dataset. Note that for each dataset the trained model with cutoff value in **bold** is the one chosen for comparison in Chapter 4.

MSRA corpus:

cutoff	prec.	recall	F1	Loc. F1	Org. F1	Per. F1
$c = 1$	93.00%	85.45%	89.07	90.95	83.73	91.57
$c = \mathbf{2}$	92.87%	86.01%	**89.31**	91.05	83.86	92.23
$c = 3$	92.56%	86.00%	89.16	90.89	83.63	92.17

CityU corpus:

cutoff	prec.	recall	F1	Loc. F1	Org. F1	Per. F1
$c = 3$	90.08%	72.51%	80.34	86.69	66.96	81.44
$c = \mathbf{4}$	89.76%	72.88%	**80.44**	86.67	67.14	81.63
$c = 5$	89.61%	72.93%	80.41	86.58	66.93	81.77

PDJ98 corpus:

	cutoff	prec.	recall	F1	Loc. F1	Org. F1	Per. F1
fold 1	$c = \mathbf{2}$	91.63%	82.56%	**86.86**	86.55	84.58	88.87
	$c = 3$	91.44%	82.63%	86.81	86.28	84.39	89.26
	$c = 4$	91.24%	82.73%	86.78	86.27	84.19	89.29
fold 2	$c = 2$	91.63%	81.41%	86.22	86.96	83.79	86.83
	$c = \mathbf{3}$	91.49%	81.68%	**86.31**	87.09	83.66	87.00
	$c = 4$	91.38%	81.70%	86.27	87.00	83.69	87.00
fold 3	$c = \mathbf{2}$	93.24%	84.98%	**88.92**	88.64	87.57	90.13
	$c = 3$	92.97%	84.97%	88.79	88.46	87.16	90.24
	$c = 4$	92.74%	84.92%	88.66	88.26	87.15	90.12
fold 4	$c = \mathbf{2}$	93.89%	85.74%	**89.63**	88.92	86.31	91.73
	$c = 3$	93.63%	85.90%	89.60	88.87	86.05	91.81
	$c = 4$	93.37%	85.65%	89.34	88.44	85.92	91.68
fold 5	$c = 2$	91.45%	81.31%	86.08	85.71	87.04	85.98
	$c = 3$	91.22%	81.41%	86.04	85.60	86.99	86.05
	$c = \mathbf{4}$	91.13%	81.62%	**86.11**	85.45	87.20	86.36

Note that we have not tried the case that $c = 1$, i.e. no feature cutoff effectively, for all the 5 folds of PKJ98, as this is well-known to induce over-training.

Appendix B

This appendix shows the details of the paired t-tests about comparing the various performance metrics of CRFDP-basic and CRFDP-full *on the PDJ98 corpus.*

Paired t-test on comparing the overall F1-scores of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on the overall F1-score.

fold	F1-score		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	86.55	87.44	0.89
2	86.26	86.95	0.69
3	88.83	89.47	0.64
4	89.06	90.37	1.31
5	85.71	86.68	0.97

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.90 \\ \text{std. dev. of difference } s_d &= 0.267 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 7.54$$

On 4 degrees of freedom, the p-value corresponding to t is 8.28×10^{-4}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on the overall F1-score.

Paired t-test on comparing the overall recall of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on the overall recall.

	Recall		difference
fold	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	82.78	84.07	1.29
2	82.13	83.30	1.17
3	85.44	86.22	0.78
4	85.74	87.44	1.70
5	81.51	82.84	1.33

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 1.25 \\ \text{std. dev. of difference } s_d &= 0.15 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 8.48$$

On 4 degrees of freedom, the p-value corresponding to t is 5.31×10^{-4}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on the overall recall.

Paired t-test on comparing the overall precision of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on the overall precision.

fold	Precision		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	90.68	91.10	0.42
2	90.82	90.94	0.12
3	92.50	92.97	0.47
4	92.65	93.49	0.84
5	90.35	90.90	0.55

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.48 \\ \text{std. dev. of difference } s_d &= 0.116 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 4.15$$

On 4 degrees of freedom, the p-value corresponding to t is 7.14×10^{-3}. Therefore, at a significance level of 1%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on the overall precision.

Paired t-test on comparing the location F1-scores of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on location F1-score.

fold	F1-score		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	86.09	87.19	1.10
2	86.36	87.24	0.88
3	88.44	89.21	0.77
4	88.40	89.40	1.00
5	85.47	86.31	0.84

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.92 \\ \text{std. dev. of difference } s_d &= 0.13 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 15.60$$

On 4 degrees of freedom, the p-value corresponding to t is 4.93×10^{-5}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on location F1-score.

Paired t-test on comparing the organization F1-scores of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on organization F1-score.

fold	F1-score		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	83.43	83.90	0.47
2	83.31	84.07	0.76
3	87.32	87.53	0.21
4	84.41	85.98	1.57
5	85.71	86.34	0.63

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.73 \\ \text{std. dev. of difference } s_d &= 0.51 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 3.17$$

On 4 degrees of freedom, the p-value corresponding to t is 1.69×10^{-2}. Therefore, at a significance level of 2%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on organization F1-score.

Paired t-test on comparing the person F1-scores of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on person F1-score.

	F1-score		difference
fold	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	89.31	90.14	0.83
2	88.16	88.56	0.40
3	90.27	90.98	0.71
4	91.66	93.12	1.46
5	86.06	87.46	1.40

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.96 \\ \text{std. dev. of difference } s_d &= 0.46 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 4.69$$

On 4 degrees of freedom, the p-value corresponding to t is 4.68×10^{-3}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on person F1-score.

Paired t-test on comparing the location recall of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on location recall.

	Recall		difference
fold	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	82.87	84.58	1.71
2	83.91	85.11	1.20
3	85.75	86.53	0.78
4	85.87	86.75	0.88
5	82.37	83.57	1.20

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 1.15 \\ \text{std. dev. of difference } s_d &= 0.36 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 7.10$$

On 4 degrees of freedom, the p-value corresponding to t is 1.04×10^{-3}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on location recall.

Paired t-test on comparing the organization recall of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on organization recall.

fold	Recall		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	79.31	79.50	0.19
2	78.61	79.95	1.34
3	84.21	84.48	0.27
4	80.85	82.53	1.68
5	81.45	81.83	0.38

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.77 \\ \text{std. dev. of difference } s_d &= 0.69 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 2.51$$

On 4 degrees of freedom, the p-value corresponding to t is 3.30×10^{-2}. Therefore, at a significance level of 5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on organization recall.

Paired t-test on comparing the person recall of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on person recall.

fold	Recall		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	84.92	86.30	1.38
2	82.00	83.02	1.02
3	85.76	86.85	1.09
4	87.66	90.17	2.51
5	80.32	82.44	2.12

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 1.62 \\ \text{std. dev. of difference } s_d &= 0.66 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 5.51$$

On 4 degrees of freedom, the p-value corresponding to t is 2.66×10^{-3}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on person recall.

Paired t-test on comparing the location precision of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on location precision.

fold	Precision		difference
	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	89.57	89.97	0.40
2	88.96	89.47	0.51
3	91.30	92.06	0.76
4	91.08	92.22	1.14
5	88.81	89.22	0.41

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.64 \\ \text{std. dev. of difference } s_d &= 0.31 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 4.60$$

On 4 degrees of freedom, the p-value corresponding to t is 5.01×10^{-3}. Therefore, at a significance level of 1%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on location precision.

Paired t-test on comparing the organization precision of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on organization precision.

	Precision		difference
fold	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	88.00	88.82	0.82
2	88.60	88.65	0.05
3	90.66	90.82	0.16
4	88.29	89.74	1.45
5	90.43	91.37	0.94

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.684 \\ \text{std. dev. of difference } s_d &= 0.58 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 2.64$$

On 4 degrees of freedom, the p-value corresponding to t is 2.89×10^{-2}. Therefore, at a significance level of 5%, it can be concluded that CRFDP-full on average does have improvement over CRFDP-basic on organization precision.

Paired t-test on comparing the person precision of CRFDP-basic and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over CRFDP-basic on person precision.

	Precision		difference
fold	CRFDP-basic (x)	CRFDP-full (y)	($d = y - x$)
1	94.18	94.33	0.14
2	95.32	94.89	-0.43
3	95.29	95.54	0.25
4	96.04	96.27	0.23
5	92.69	93.13	0.44

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.13 \\ \text{std. dev. of difference } s_d &= 0.33 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 0.87$$

On 4 degrees of freedom, the p-value corresponding to t is 0.22. Therefore, at a significance level of 5%, it cannot be concluded that CRFDP-full on average has improvement over CRFDP-basic on person precision.

Appendix C

This appendix shows the details of the paired t-tests about comparing the various performance metrics of OrigCRF and CRFDP-full *on the PDJ98 corpus.*

Paired t-test on comparing the overall F1-scores of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on the overall F1-score.

	F1-score		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	86.86	87.44	0.58
2	86.31	86.95	0.64
3	88.92	89.47	0.55
4	89.63	90.37	0.74
5	86.11	86.68	0.57

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.62 \\ \text{std. dev. of difference } s_d &= 0.077 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 17.89$$

On 4 degrees of freedom, the p-value corresponding to t is 2.87×10^{-5}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on the overall F1-score.

Paired t-test on comparing the overall recall of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on the overall recall.

fold	Recall		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	82.56	84.07	1.51
2	81.68	83.30	1.62
3	84.98	86.22	1.24
4	85.74	87.44	1.70
5	81.62	82.84	1.22

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 1.46 \\ \text{std. dev. of difference } s_d &= 0.22 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 14.89$$

On 4 degrees of freedom, the p-value corresponding to t is 5.92×10^{-5}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on the overall recall.

Paired t-test on comparing the overall precision of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on the overall precision.

	Precision		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	91.63	91.10	-0.53
2	91.49	90.94	-0.55
3	93.24	92.97	-0.27
4	93.89	93.49	-0.40
5	91.13	90.90	-0.23

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= -0.40 \\ \text{std. dev. of difference } s_d &= 0.065 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -6.07$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on the overall precision.

Paired t-test on comparing the location F1-scores of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on location F1-score.

fold	F1-score		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	86.55	87.19	0.64
2	87.09	87.24	0.15
3	88.64	89.21	0.57
4	88.92	89.40	0.48
5	85.45	86.31	0.86

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 0.54 \\ \text{std. dev. of difference } s_d &= 0.26 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 4.66$$

On 4 degrees of freedom, the p-value corresponding to t is 4.81×10^{-3}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on location F1-score.

Paired t-test on comparing the organization F1-scores of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on organization F1-score.

fold	F1-score		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	84.58	83.90	-0.68
2	83.66	84.07	0.41
3	87.57	87.53	-0.04
4	86.31	85.98	-0.33
5	87.20	86.34	-0.86

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n-1=4 \\ \text{mean difference } \bar{d} &= -0.30 \\ \text{std. dev. of difference } s_d &= 0.51 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -1.32$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on organization F1-score.

Paired t-test on comparing the person F1-scores of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on person F1-score.

	F1-score		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	88.87	90.14	1.27
2	87.00	88.56	1.56
3	90.13	90.98	0.85
4	91.73	93.12	1.39
5	86.36	87.46	1.10

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n-1=4 \\ \text{mean difference } \bar{d} &= 1.23 \\ \text{std. dev. of difference } s_d &= 0.12 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 10.12$$

On 4 degrees of freedom, the p-value corresponding to t is 2.68×10^{-4}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on person F1-score.

Paired t-test on comparing the location recall of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on location recall.

	Recall		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	82.56	84.58	2.02
2	84.18	85.11	0.93
3	85.13	86.53	1.40
4	85.73	86.75	1.02
5	81.65	83.57	1.92

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 1.46 \\ \text{std. dev. of difference } s_d &= 0.50 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 6.51$$

On 4 degrees of freedom, the p-value corresponding to t is 1.44×10^{-3}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on location recall.

Paired t-test on comparing the organization recall of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on organization recall.

fold	Recall		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	80.89	79.50	-1.39
2	78.96	79.95	0.99
3	84.92	84.48	-0.44
4	82.64	82.53	-0.11
5	83.53	81.83	-1.70

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= -0.53 \\ \text{std. dev. of difference } s_d &= 1.07 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -1.10$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on organization recall.

Paired t-test on comparing the person recall of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on person recall.

fold	Recall		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	83.66	86.30	2.64
2	79.95	83.02	3.07
3	84.83	86.85	2.02
4	87.05	90.17	3.12
5	80.32	82.44	2.12

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= 2.59 \\ \text{std. dev. of difference } s_d &= 0.51 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = 11.27$$

On 4 degrees of freedom, the p-value corresponding to t is 1.77×10^{-4}. Therefore, at a significance level of 0.5%, it can be concluded that CRFDP-full on average does have improvement over OrigCRF on person recall.

Paired t-test on comparing the location precision of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on location precision.

fold	Precision		difference
	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	90.93	89.97	-0.96
2	90.22	89.47	-0.75
3	92.45	92.06	-0.39
4	92.35	92.22	-0.13
5	89.62	89.22	-0.40

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n-1=4 \\ \text{mean difference } \bar{d} &= -0.53 \\ \text{std. dev. of difference } s_d &= 0.33 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -3.59$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on location precision.

Paired t-test on comparing the organization precision of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on organization precision.

	Precision		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	88.62	88.82	0.20
2	88.95	88.65	-0.30
3	90.39	90.82	0.43
4	90.32	89.74	-0.58
5	91.20	91.37	0.17

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n-1=4 \\ \text{mean difference } \bar{d} &= -0.016 \\ \text{std. dev. of difference } s_d &= 0.41 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -0.09$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on organization precision.

Paired t-test on comparing the person precision of OrigCRF and CRFDP-full

Null hypothesis: CRFDP-full does not have improvement over OrigCRF on person precision.

	Precision		difference
fold	orig.CRF (x)	CRFDP-full (y)	($d = y - x$)
1	94.77	94.33	-0.44
2	95.41	94.89	-0.52
3	96.12	95.54	-0.58
4	96.93	96.27	-0.66
5	93.38	93.13	-0.25

$$\begin{aligned} \text{no. of folds } n &= 5 \\ \text{degrees of freedom} &= n - 1 = 4 \\ \text{mean difference } \bar{d} &= -0.49 \\ \text{std. dev. of difference } s_d &= 0.16 \end{aligned}$$

So, we have

$$t = \frac{\bar{d} \times \sqrt{n}}{s_d} = -7.00$$

For t being negative, it is not possible to have a p-value lower than 0.5%. Therefore, at a significance level of 0.5%, it cannot be concluded that CRFDP-full on average has improvement over OrigCRF on person precision.