A Graph-based Approach to Named Entity Categorization in Wikipedia Using Conditional Random Fields

Yotaro Watanabe, Masayuki Asahara and Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
{yotaro-w,masayu-a,matsu}@is.naist.jp

Abstract

This paper presents a method for categorizing named entities in Wikipedia. In Wikipedia, an anchor text is glossed in a linked HTML text. We formalize named entity categorization as a task of categorizing anchor texts with linked HTML texts which glosses a named entity. Using this representation, we introduce a graph structure in which anchor texts are regarded as nodes. In order to incorporate HTML structure on the graph, three types of cliques are defined based on the HTML tree structure. We propose a method with Conditional Random Fields (CRFs) to categorize the nodes on the graph. Since the defined graph may include cycles, the exact inference of CRFs is computationally expensive. We introduce an approximate inference method using Tree-based Reparameterization (TRP) to reduce computational cost. In experiments, our proposed model obtained significant improvements compare to baseline models that use Support Vector Machines.

1 Introduction

Named and Numeric Entities (NEs) refer to proper nouns (e.g. PERSON, LOCATION and ORGANIZATION), time expressions, date expressions and so on. Since a large number of NEs exist in the world, unknown expressions appear frequently in texts, and they become hindrance to real-world text analysis. To cope with the problem, one effective ways to add a large number of NEs to gazetteers.

In recent years, NE extraction has been performed with machine learning based methods. However, such methods cannot cover all of NEs in texts. Therefore, it is necessary to extract NEs from existing resources and use them to identify more NEs. There are many useful resources on the Web. We focus on Wikipedia[1] as the resource for acquiring NEs. Wikipedia is a free multilingual online encyclopedia and a rapidly growing resource. In Wikipedia, a large number of NEs are described in titles of articles with useful information such as HTML tree structures and categories. Each article links to other related articles. According to these characteristics, they could be an appropriate resource for extracting NEs.

Since a specific entity or concept is glossed in a Wikipedia article, we can regard the NE extraction problem as a document classification problem of the Wikipedia article. In traditional approaches for document classification, in many cases, documents are classified independently. However, the Wikipedia articles are hypertexts and they have a rich structure that is useful for categorization. For example, hyperlinked mentions (we call them **anchor texts**) which are enumerated in a list tend to refer to the articles that describe other NEs belonging to the same class. It is expected that improved NE categorization is accomplished by capturing such dependencies.

We structure anchor texts and dependencies between them into a graph, and train graph-based CRFs to obtain probabilistic models to estimate categories for NEs in Wikipedia.

So far, several statistical models that can cap-

[1] http://wikipedia.org/

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 649–657, Prague, June 2007.

ture dependencies between examples have been proposed. There are two types of classification methods that can capture dependencies: iterative classification methods (Neville and Jensen, 2000; Lu and Getoor, 2003b) and collective classification methods (Getoor et al., 2001; Taskar et al., 2002). In this paper, we use Conditional Random Fields (CRFs) (Lafferty et al., 2001) for NE categorization in Wikipedia.

The rest of the paper is structured as follows. Section 2 describes the general framework of CRFs. Section 3 describes a graph-based CRFs for NE categorization in Wikipedia. In section 4, we show the experimental results. Section 5 describes related work. We conclude in section 6.

2 Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are undirected graphical models that give a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ in a form of exponential model.

CRFs are formalized as follows. Let $\mathcal{G} = \{V, E\}$ be an undirected graph over random variables $\boldsymbol{y}$ and $\boldsymbol{x}$, where V is a set of vertices, and E is a set of edges in the graph $\mathcal{G}$. When a set of cliques $C = \{\{\boldsymbol{y}_c, \boldsymbol{x}_c\}\}$ are given, CRFs define the conditional probability of a state assignment given an observation set.

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{c \in \boldsymbol{C}} \Phi(\boldsymbol{x}_c, \boldsymbol{y}_c) \quad (1)$$

where $\Phi(\boldsymbol{x}_c, \boldsymbol{y}_c)$ is a potential function defined over cliques, and $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \prod_{c \in \boldsymbol{C}} \Phi(\boldsymbol{x}_c, \boldsymbol{y}_c)$ is the partition function.

The potentials are factorized according to the set of features $\{f_k\}$.

$$\Phi(\boldsymbol{x}_c, \boldsymbol{y}_c) = \exp\left(\sum_k \lambda_k f_k(\boldsymbol{x}_c, \boldsymbol{y}_c)\right) \quad (2)$$

where $F = \{f_1, ..., f_K\}$ are feature functions on the cliques, $\Lambda = \{\lambda_1, ..., \lambda_K \in \mathcal{R}\}$ are the model parameters. The parameters Λ are estimated iterative scaling or quasi-Newton method from labeled data.

The original paper (Lafferty et al., 2001) focused on linear-chain CRFs, and applied them to part-of-speech tagging problem. McCallum et al. (2003), Sutton et al (2004) proposed Dynamic Conditional Random Fields (DCRFs), the generalization of linear-chain CRFs, that have complex graph structure (include cycles). Since DCRFs model structure contains cycles, it is necessary to use approximate inference methods to calculate marginal probability. Tree-based Reparameterization (TRP) (Wainwright et al., 2003), a schedule for loopy belief propagation, is used for approximate inference in these papers.

3 Graph-based CRFs for NE Categorization in Wikipedia

In this section we describe how to apply CRFs for NE categorization in Wikipedia.

Each Wikipedia article describes a specific entity or concept by a heading word, a definition, and one or more categories. One possible approach is to classify each NE described in an article into an appropriate category by exploiting the definition of the article. This process can be done one by one without considering the relationship with other articles.

On the other hand, articles in Wikipedia are semi-structured texts. Especially lists (<UL> or <OL>) and tables (<TABLE>) have an important characteristics, that is, occurrence of elements in them have some sort of dependencies. Structural characteristics, such as lists (<UL> or <OL>) or tables (<TABLE>), are useful becase their elements have some sort of dependencies.

Figure 2 shows an example of an HTML segment and the corresponding tree structure. The first anchor texts in each list tag (<LI>) tend to be in the same NE category. Such characteristics are useful feature for the categorization task. In this paper we focus on lists which appear frequently in Wikipedia.

Furthermore, there are anchor texts in articles. Anchor texts are glossed entity or concept described with links to other pages. With this in mind, our NE categorization problem can be regarded as NE category labeling problem for anchor texts in articles. Exploiting dependencies of anchor texts that are induced by the HTML structure is expected to improve categorization performance.

We use CRFs for categorization in which anchor texts correspond to random variables V in $\mathcal{G}$ and de-

Sibling $E_S = \{(v_i^{\mathcal{T}}, v_j^{\mathcal{T}}) | v_i^{\mathcal{T}}, v_j^{\mathcal{T}} \in V^{\mathcal{T}},\ d(v_i^{\mathcal{T}},\ ca(v_i^{\mathcal{T}},\ v_j^{\mathcal{T}})) = d(v_j^{\mathcal{T}},\ ca(v_i^{\mathcal{T}},\ v_j^{\mathcal{T}})) = 1,\ v_j^{\mathcal{T}} = ch(pa(v_j^{\mathcal{T}}, 1), k),$
$v_i^{\mathcal{T}} = ch(pa(v_i^{\mathcal{T}}, 1),\ \max\{l | l < k\})\}$

Cousin $E_C = \{(v_i^{\mathcal{T}}, v_j^{\mathcal{T}}) | v_i^{\mathcal{T}},\ v_j^{\mathcal{T}} \in V^{\mathcal{T}}, d(v_i^{\mathcal{T}},\ ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) = d(v_j^{\mathcal{T}},\ ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) \geq 2,\ v_i^{\mathcal{T}} = ch(pa(v_i^{\mathcal{T}}), k),$
$v_j^{\mathcal{T}} = ch(pa(v_j^{\mathcal{T}}), k),\ pa(v_j^{\mathcal{T}}, d(v_j^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) - 1) = ch(pa(v_j^{\mathcal{T}}, d(v_j^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}}))), k),$
$pa(v_i^{\mathcal{T}}, d(v_i^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) - 1) = ch(pa(v_i^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})), \max\{l | l < k\})\}$

Relative $E_R = \{(v_i^{\mathcal{T}}, v_j^{\mathcal{T}}) | v_i^{\mathcal{T}}, v_j^{\mathcal{T}} \in V^{\mathcal{T}}, d(v_i^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) = 1, d(v_j^{\mathcal{T}}, ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})) = 3,$
$pa(v_j^{\mathcal{T}}, 2) = ch(pa(v_j^{\mathcal{T}}, 3), k), v_i^{\mathcal{T}} = ch(pa(v_i^{\mathcal{T}}, 1), \max\{l | l < k\})\}$

Figure 1: The definitions of sibling, cousin and relative cliques, where E_S, E_C, E_R correspond to sets which consist of anchor text pairs that have sibling, cousin and relative relations respectively.

pendencies between anchor texts are treated as edges E in $\mathcal{G}$. In the next section, we describe the concrete way to construct graphs.

3.1 Constructing a graph from an HTML tree

An HTML document is an ordered tree. We define a graph $\mathcal{G} = (V^{\mathcal{G}}, E^{\mathcal{G}})$ on an HTML tree $\mathcal{T}^{HTML} = (V^{\mathcal{T}}, E^{\mathcal{T}})$: the vertices $V^{\mathcal{G}}$ are anchor texts in the HTML text; the edges E are limited to cliques of Sibling, Cousin, and Relative, which we will describe later in the section. These cliques are intended to encode a NE label dependency between anchor texts where the two NEs tend to be in the same or related class, or one NE affects the other NE label.

Let us consider dependent anchor text pairs in Figure 2. First, "Dillard & Clark" and "country rock" have a sibling relation over the tree structure, and appearing the same element of the list. The latter element in this relation tends to be an attribute or a concept of the other element in the relation. Second, "Dillard & Clark" and "Carpenters" have a cousin relation over the tree structure, and they tend to have a common attribute such as "Artist". The elements in this relation tend to belong to the same class. Third, "Carpenters" and "Karen Carpenter" have a relation in which "Karen Carpenter" is a sibling's grandchild in relation to "Carpenters" over the tree structure. The latter elements in this relation tends to be a constituent part of the other element in the relation. We can say that the model can capture dependencies by dealing with anchor texts that depend on each other as cliques. Based on the observations as above, we treat a pair of anchor texts as cliques which satisfy the condtions in Figure 1.

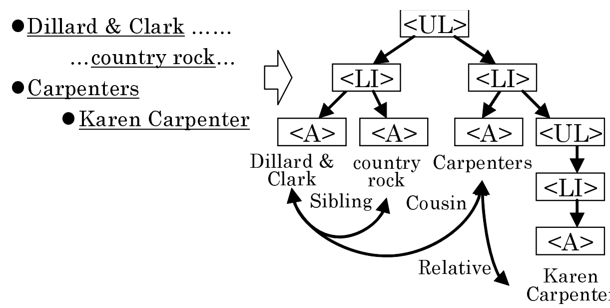

Figure 2: Correspondence between tree structure and defined cliques.

Now, we define the three sorts of edges given an HTML tree. Consider an HTML tree $\mathcal{T}^{HTML} = (V^{\mathcal{T}}, E^{\mathcal{T}})$, where $V^{\mathcal{T}}$ and $E^{\mathcal{T}}$ are nodes and edges over the tree. Let $d(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})$ be the number of edges between $v_i^{\mathcal{T}}$ and $v_j^{\mathcal{T}}$ where $v_i^{\mathcal{T}}, v_j^{\mathcal{T}} \in V^{\mathcal{T}}$, $pa(v_i^{\mathcal{T}}, k)$ be k-th generation ancestor of $v_i^{\mathcal{T}}$, $ch(v_i^{\mathcal{T}}, k)$ be $v_i^{\mathcal{T}}$'s k-th child, $ca(v_i^{\mathcal{T}}, v_j^{\mathcal{T}})$ be a common ancestor of $v_i^{\mathcal{T}}, v_j^{\mathcal{T}} \in V^{\mathcal{T}}$. Precise definitions of cliques, namely Sibling, Cousin, and Relative, are given in Figure 1. A set of cliques used in our graph-based CRFs are edges defined in Figure 1 and vertices, i.e. $C = E_S \cup E_C \cup E_R \cup V$. Note that they are restricted to pairs of the nearest vertices to keep the graph simple.

3.2 Model

We introduce potential functions for cliques to define conditional probability distribution over CRFs. Conditional distribution over label set $\boldsymbol{y}$ given observation set $\boldsymbol{x}$ is defined as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \left(\prod_{(v_i,v_j)\in E_S \cup E_C \cup E_R} \Phi_{SCR}(y_i, y_j) \right) \left(\prod_{v_i \in V} \Phi_V(y_i, \boldsymbol{x}) \right) \quad (3)$$

where $\Phi_{SCR}(y_i, y_j)$ is the potential over sibling, cousin and relative edges, $\Phi_V(y_i, \boldsymbol{x})$ is the potential over the nodes, and $Z(\boldsymbol{x})$ is the partition function. The potentials $\Phi_{SCR}(y_i, y_j)$ and $\Phi_V(y_i, \boldsymbol{x})$ factorize according to the features f_k and weights λ_k as:

$$\Phi_{SCR}(y_i, y_j) = \exp\left(\sum_k \lambda_k f_k(y_i, y_j) \right) \quad (4)$$

$$\Phi_V(y_i, \boldsymbol{x}) = \exp\left(\sum_{k'} \lambda_{k'} f_{k'}(y_i, \boldsymbol{x}) \right) \quad (5)$$

$f_k(y_i, y_j)$ captures co-occurrences between labels, where $k \in \{(y_i, y_j)|\mathcal{Y} \times \mathcal{Y}\}$ corresponds to the particular element of the Cartesian product of the label set $\mathcal{Y}$. $f_{k'}(y_i, \boldsymbol{x})$ captures co-occurrences between label $y_i \in \mathcal{Y}$ and observation features, where k' corresponds to the particular element of the label set and observed features.

The weights of a CRF, $\Lambda = \{\lambda_k, \ldots, \lambda_{k'}, \ldots\}$ are estimated to maximize the conditional log-likelihood of the graph in a training dataset $\mathcal{D} = \{\langle \boldsymbol{x}^{(1)}, y^{(1)} \rangle, \langle \boldsymbol{x}^{(2)}, y^{(2)} \rangle, \ldots, \langle \boldsymbol{x}^{(N)}, y^{(N)} \rangle\}$ The log-likelihood function can be defined as follows:

$$\mathcal{L}_\lambda = \sum_{d=1}^{N} [\sum_{(v_i,v_j)\in E_S^{(d)} \cup E_C^{(d)} \cup E_R^{(d)}} \sum_k \lambda_k f_k(y_i, y_j) + \sum_{v_i \in V^{(d)}} \sum_{k'} \lambda_{k'} f_{k'}(y_i, \boldsymbol{x}^{(d)}) - logZ(\boldsymbol{x}^{(d)})] - \sum_k \frac{\lambda_k^2}{2\sigma^2} - \sum_{k'} \frac{\lambda_{k'}^2}{2\sigma^2} \quad (6)$$

where the last two terms are due to the Gaussian prior (Chen and Rosenfeld, 1999) used to reduce overfitting. Quasi-Newton methods, such as L-BFGS (Liu and Nocedal, 1989) can be used for maximizing the function.

3.3 Tree-based Reparameterization

Since the proposed model may include loops, it is necessary to introduce an approximation to calculate mariginal probabilities. For this, we use Tree-based Reparameterization (TRP) (Wainwright et al., 2003) for approximate inference. TRP enumerates a set of spanning trees from the graph. Then, inference is performed by applying an exact inference algorithm such as Belief Propagation to each of the spanning trees, and updates of marginal probabilities are continued until they converge.

4 Experiments

4.1 Dataset

Our dataset is a random selection of 2300 articles from the Japanese version of Wikipedia as of October 2005. All anchor texts appearing under HTML <LI> tags are hand-annotated with NE class label. We use the Extended Named Entity Hierarchy (Sekine et al., 2002) as the NE class labeling guideline, but reduce the number of classes to 13 from the original 200+ by ignoring fine-grained categories and nearby categories in order to avoid data sparseness. We eliminate examples that consist of less than two nodes in the SCR model. There are 16136 anchor texts with 14285 NEs. The number of Sibling, Cousin and Relative edges in the dataset are $|E_S| = 4925$, $|E_C| = 13134$ and $|E_R| = 746$ respectively.

4.2 Experimental settings

The aims of experiments are the two-fold. Firstly, we investigate the effect of each cliques. The several graphs are composed with the three sorts of edges. We also compare the graph-based models with a node-wise method – just MaxEnt method not using any edge dependency. Secondly, we compare the proposed method by CRFs with a baseline method by Support Vector Machines (SVMs) (Vapnik, 1998).

The experimental settings of CRFs and SVMs are as follows.

CRFs In order to investigate which type of clique boosts classification performance, we perform experiments on several CRFs models that are constructed from combinations of defined cliques. Re-

	SCR	SC	SR	CR
# of loopy examples	318 (36%)	324 (32%)	101 (1%)	42 (2%)
# of linear chain or tree examples	555 (64%)	631 (62%)	2883 (27%)	1464 (54%)
# of one node examples	0 (0%)	60 (6%)	7800 (72%)	1176 (44%)
# of total examples	873	1015	10784	2682
average # of nodes per example	18.5	15.8	1.5	6.0
	S	**C**	**R**	**I**
# of loopy examples	0 (0%)	0 (0%)	0 (0%)	0 (0%)
# of linear chain or tree examples	2913 (26%)	1631 (54%)	237 (2%)	0 (0%)
# of one node examples	8298 (74%)	1380 (46%)	15153 (98%)	16136 (100%)
# of total examples	11211	3011	15390	16136
average # of nodes per example	1.4	5.4	1.05	1

Table 1: The dataset details constructed from each model.

sulting models of CRFs evaluated on this experiments are SCR, SC, SR, CR, S, C, R and I (independent). Figure 3 shows representative graphs of the eight models. When the graph are disconnected by reducing the edges, the classification is performed on each connected subgraph. We call it an *example*. We name the *examples* according the graph structure: ”loopy examples” are subgraphs including at least one cycle; ”linear chain or tree examples” are subgraphs including not a cycle but at least an edge; ”one node examples” are subgraphs without edges. Table 1 shows the distribution of the examples of each model. Since SCR, SC, SR and CR model have loopy examples, TRP approximate inference is necessary. To perform training and testing with CRFs, we use GRMM (Sutton, 2006) with TRP. We set the Gaussian Prior variances for weights as $\sigma^2 = 10$ in all models.

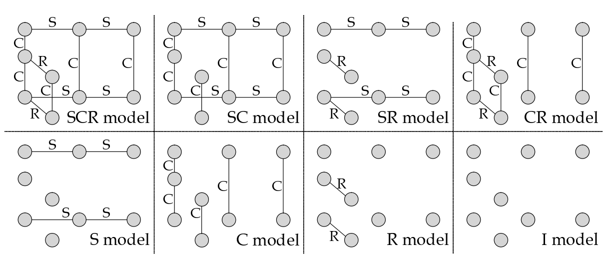

Figure 3: An example of graphs constructed by combination of defined cliques. S, C, R in the model names mean that corresponding model has Sibling, Cousin, Relative cliques respectively. In each model, classification is performed on each connected subgraph.

SVMs We introduce two models by SVMs (model I and model P). In model I, each anchor text is classified independently. In model P, we ordered the anchor texts in a linear-chain sequence. Then, we perform a history-based classification along the sequence, in which $j-1$-th classification result is used in j-th classification. We use TinySVM with a linear-kernel. One-versus-rest method is used for multi-class classification. To perform training and testing with SVMs, we use TinySVM [2] with a linear-kernel, and one-versus-rest is used for multi-class classification. We used the cost of constraint violation $C = 1$.

Features for CRFs and SVMs The features used in the classification with CRFs and SVMs are shown in Table 2. Japanese morphological analyzer MeCab [3] is used to obtain morphemes.

4.3 Evaluation

We evaluate the models by 5 fold cross-validation. Since the number of examples are different in each model, the datasets are divided taking the examples – namely, connected subgraphs – in SCR model. The size of divided five sub-data are roughly equal. We evaluate per-class and total extraction performance by F1-value.

4.4 Results and discussion

Table 3 shows the classification accuracy of each model. The second column “N” stands for the number of nodes in the gold data. The second last row “ALL” stands for the F1-value of all NE classes.

[2] http://www.chasen.org/~taku/software/TinySVM/

[3] http://mecab.sourceforge.net/

types	feature	SVMs	CRFs
observation features	definition (bag-of-words)	$\surd$	$\surd\,(V)$
	heading of articles	$\surd$	$\surd\,(V)$
	heading of articles (morphemes)	$\surd$	$\surd\,(V)$
	categories articles	$\surd$	$\surd\,(V)$
	categories articles (morphemes)	$\surd$	$\surd\,(V)$
	anchor texts	$\surd$	$\surd\,(V)$
	anchor texts (morphemes)	$\surd$	$\surd\,(V)$
	parent tags of anchor texts	$\surd$	$\surd\,(V)$
	text included in the last header of anchor texts	$\surd$	$\surd\,(V)$
	text included in the last header of anchor texts(morphemes)	$\surd$	$\surd\,(V)$
label features	between-label feature		$\surd\,(S, C, R)$
	previous label	$\surd$	

Table 2: Features used in experiments. "$\surd$" means that the corresponding features are used in classification. The V, S, C and R in CRFs column corresponds to the node, sibling edges, cousin edges and relative edges respectively.

		CRFs								SVMs	
NE CLASS	N	C	CR	I	R	S	SC	SCR	SR	I	P
PERSON	3315	.7419	.7429	.7453	.7458	.7507	.7533	**.7981**	.7515	.7383	.7386
TIMEX/NUMEX	2749	.9936	**.9944**	.9940	.9936	.9938	.9931	.9933	.9940	.9933	.9935
FACILITY	2449	.8546	.8541	.8540	.8516	.8500	.8530	.8495	.8495	.8504	**.8560**
PRODUCT	1664	.7414	**.7540**	.7164	.7208	.7130	.7371	.7418	.7187	.7154	.7135
LOCATION	1480	**.7265**	.7239	.6989	.7048	.6974	.7210	.7232	.7033	.7022	.7132
NATURAL_OBJECTS	1132	.3333	.3422	.3476	.3513	.3547	.3294	.3304	.3316	**.3670**	.3326
ORGANIZATION	991	.7122	.7160	.7100	.7073	.7122	.6961	.5580	.7109	.7141	**.7180**
VOCATION	303	.9088	.9050	.9075	.9059	.9150	.9122	.9100	**.9186**	.9091	.9069
EVENT	121	.2740	.2345	.2533	.2667	.2800	.2740	.2759	.2667	.3418	**.3500**
TITLE	42	.1702	.0889	.2800	.2800	**.3462**	.2083	.1277	**.3462**	.2593	.2642
NAME_OTHER	24	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	**.0690**	.0000
UNIT	15	.2353	.1250	.2353	.2353	.2353	.1250	.1250	.2353	**.3333**	.3158
ALL	14285	.7846	**.7862**	.7806	.7814	.7817	.7856	.7854	.7823	.7790	.7798
ALL (no articles)	3898	.5476	**.5495**	.5249	.5274	.5272	.5484	.5465	.5224	.5278	.5386

Table 3: Comparison of F1-values of CRFs and SVMs.

The last row "ALL (no article)" stands for the F1-value of all NE classes which have no gloss texts in Wikipedia.

Relational vs. Independent Among the models constructed by combination of defined cliques, the best F1-value is achieved by CR model, followed by SC, SCR, C, SR, S, R and I. We performed McNemar paired test on labeling disagreements between CR model of CRFs and I model of CRFs. The difference was significant ($p < 0.01$). These results show that considering dependencies work positively in obtaining better accuracy than classifying independently. The Cousin cliques provide the highest accuracy improvement among the three defined cliques. The reason may be that the Cousin cliques appear frequently in comparison with the other cliques, and also possess strong dependencies among anchor texts. As for PERSON, better accuracy is achieved in SC and SCR models. In fact, the PERSON-PERSON pairs frequently appear in Sibling cliques (435 out of 4925) and in Cousin cliques (2557 out of 13125) in the dataset. Also, as for PRODUCT and LOCATION, better accuracy is achieved in the models that contain Cousin cliques (C, CR, SC and SCR model). 1072 PRODUCT-PRODUCT pairs and 738 LOCATION-LOCATION pairs appear in Cousin cliques. "All (no article)" row in Table 3 shows the F1-value of nodes which have no gloss texts. The F1-value difference between CR and I model of CRF in "ALL (no article)" row is larger than the difference in "All" row. The fact means that the dependency information helps to extract NEs without gloss texts in Wikipedia. We attempted a different parameter tying in which the SCR potential functions are tied with a particular observation feature. This parameter tying is introduced by Ghamrawi and McCallum (2005). However, we did not get any improved accuracy.

CRFs vs. SVMs The best model of CRFs (CR model) outperforms the best model of SVMs (P model). We performed McNemar paired test on labeling disagreements between CR model of CRFs and P model of SVMs. The difference was significant ($p < 0.01$). In the classes having larger number of examples, models of CRFs achieve better F1-values than models of SVMs. However, in several classes having smaller number of examples such as

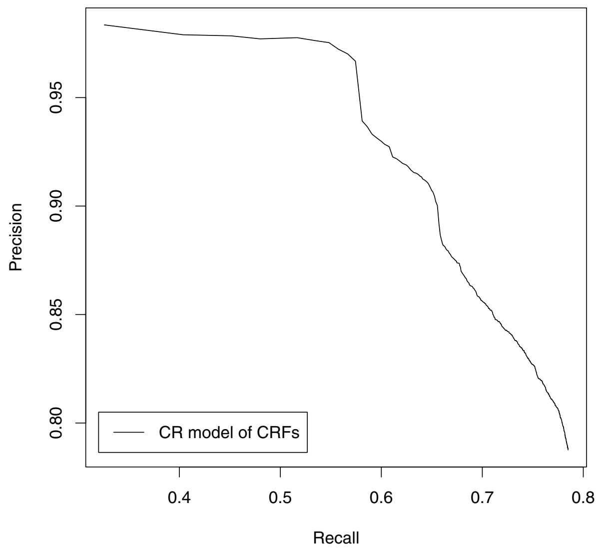

Figure 4: Precision-Recall curve obtained by varying the threshold τ of marginal probability from 1.0 to 0.0.

EVENT and UNIT, models of SVMs achieve significantly better F1-values than models of CRFs.

Filtering NE Candidates using Marginal Probability The precision-recall curve obtained by thresholding the marginal probability of the MAP estimation in the CR models is shown in Figure 4. The curve reaches a peak at 0.57 in recall, and the precision value at that point is 0.97. This precision and recall values mean that 57% of all NEs can be classified with approximately 97% accuracy on a particular thresholding of marginal probability. This results suggest that the extracted NE candidates can be filtered with fewer cost by exploiting the marginal probability.

Training Time The total training times of all CRFs and SVMs models are shown in Table 4. The training time tends to increase in case models have complicated graph structure. For instance, model SCR has complex graph structure compare to model I, therefore the SCR's training time is three times longer than model I. Training the models by SVMs are faster than training the models by CRFs. The difference comes from the implementation issues: C++

	CRFs								SVMs	
	C	CR	I	R	S	SC	SCR	SR	I	P
Training Time (minutes)	207	255	97	90	138	305	316	157	28	29

Table 4: Training Time (minutes)

vs. Java, differences of feature extraction modules, and so on. So, the comparing these two is not the important issue in this experiment.

5 Related Work

Wikipedia has become a popular resource for NLP. Bunescu and Pasca used Wikipedia for detecting and disambiguating NEs in open domain texts (2006). Strube and Ponzetto explored the use of Wikipedia for measuring Semantic Relatedness between two concepts (2006), and for Coreference Resolution (2006).

Several CRFs have been explored for information extraction from the web. Tang et al. proposed Tree-structured Conditional Random Fields (TCRFs) (2006) that capture hierarchical structure of web documents. Zhu et al. proposed Hierarchical Conditional Random Fields (HCRFs) (2006) for product information extraction from Web documents. TCRFs and HCRFs are similar to our approach described in section 4 in that the model structure is induced by page structure. However, the model structures of these models are different from our model.

There are statistical models that capture dependencies between examples. There are two types of classification approaches: iterative (Lu and Getoor, 2003b; Lu and Getoor, 2003a) or collective (Getoor et al., 2001; Taskar et al., 2002). Lu et al. (2003a; 2003b) proposed link-based classification method based on logistic regression. This model iterates local classification until label assignments converge. The results vary from the ordering strategy of local classification. In contrast to iterative classification methods, collective classification methods directly estimate most likely assignments. Getoor et al. proposed Probabilistic Relational Models (PRMs) (2001) which are built upon Bayesian Networks. Since Bayesian Networks are directed graphical models, PRMs cannot model directly the cases where instantiated graph contains cycles. Taskar et al. proposed Relational Markov Networks (RMNs) (2002). RMNs are the special case of Conditional Markov Networks (or Conditional Random Fields) in which graph structure and parameter tying are determined by SQL-like form.

As for the marginal probability to use as a confidence measure shown in Figure 4, Peng et al. (2004) has applied linear-chain CRFs to Chinese word segmentation. It is calculated by constrained forward-backward algorithm (Culotta and McCallum, 2004), and confident segments are added to the dictionary in order to improve segmentation accuracy.

6 Conclusion

In this paper, we proposed a method for categorizing NEs in Wikipedia. We defined three types of cliques that are constitute dependent anchor texts in construct CRFs graph structure, and introduced potential functions for them to reflect classification. The experimental results show that the effectiveness of capturing dependencies, and proposed CRFs model can achieve significant improvements compare to baseline methods with SVMs. The results also show that the dependency information from the HTML tree helps to categorize entities without gloss texts in Wikipedia. The marginal probability of MAP assignments can be used as confidence measure of the entity categorization. We can control the precision by filtering the confidence measure as PR curve in Figure 4. The measure can be also used as a confidence estimator in active learning in CRFs (Kim et al., 2006), where examples with the most uncertainty are selected for presentation to human annotators.

In future research, we plan to explore NE categorization with more fine-grained label set. For NLP applications such as QA, NE dictionary with fine-grained label sets will be a useful resource. However, generally, classification with statistical methods becomes difficult in case that the label set is large, because of the insufficient positive examples. It is an issue to be resolved in the future.

References

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

Aron Culotta and Andrew McCallum. 2004. Confidence estimation for information extraction. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Lise Getoor, Eran Segal, Ben Taskar, and Daphne Koller. 2001. Probabilistic models of text and link structure for hypertext classification. In *IJCAI Workshop on Text Learning: Beyond Supervision, 2001*.

Nadia Ghamrawi and Andrew McCallum. 2005. Collective multi-label classification. In *Fourteenth Conference on Information and Knowledge Management (CIKM)*.

Juanzi Li Jie Tang, Mingcai Hong and Bangyong Liang. 2006. Tree-structured conditional random fields for semantic annotation. In *Proceedings of 5th International Conference of Semantic Web (ISWC-06)*.

Seokhwan Kim, Yu Song, Kyungduk Kim, Jeong-Won Cha, and Gary Geunbae Lee. 2006. MMR-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL06)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Dong C. Liu and Jorge Nocedal. 1989. The limited memory BFGS methods for large scale optimization. In *Mathematical Programming 45*.

Qing Lu and Lise Getoor. 2003a. Link-based classification using labeled and unlabeled data. In *Proceedings of the International Conference On Machine Learning*, Washington DC, August.

Qing Lu and Lise Getoor. 2003b. Link-based text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Andrew McCallum, Khashayar Rohanimanesh, and Charles Sutton. 2003. Dynamic conditional random fields for jointly labeling multiple sequences. In *NIPS Workshop on Syntax, Semantics, and Statistics*, December.

J. Neville and D. Jensen. 2000. Iterative classification in relational data. In *Proceedings of AAAI-2000 Workshop on Learning Statistical Models from Relational Data, pages 13–20. AAAI Press*.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING)*.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of the LREC-2002*.

Michael Strube and Simone Paolo Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*.

Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the 21th International Conference on Machine Learning*.

Charles Sutton. 2006. GRMM: A graphical models toolkit. http://mallet.cs.umass.edu.

Ben Taskar, Pieter Abbeel, and Daphne Koller. 2002. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. Wiley Interscience.

Martin Wainwright, Tommi Jaakkola, and Alan Willsky. 2003. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, 45(9):1120–1146.

Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2006. Simultaneous record detection and attribute labeling in web data extraction. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.