Web Entity Detection for Semi-structured Text Data Records with Unlabeled Data[*]

Chunliang Lu[1], Lidong Bing[1], Wai Lam[1], Ki Chan[2], and Yuan Gu[1]

[1] The Chinese University of Hong Kong, Hong Kong
{cllu,ldbing,wlam,yuangu}@se.cuhk.edu.hk
[2] Hong Kong University of Science and Technology, Hong Kong
kccecia@cse.ust.hk

Abstract. We propose a framework for named entity detection from Web content associated with semi-structured text data records, by exploiting the inherent structure via a transformation process facilitating collective detection. To learn the sequential classification model, our framework does not require training labels on the data records. Instead, we make use of existing named entity repositories such as DBpedia. We incorporate this external clue via distant supervision, by making use of the Generalized Expectation constraint. After that, a collective detection model based on logical inference is proposed to consider the consistency among potential named entities as well as header text. Extensive experiments have been conducted to evaluate the effectiveness of our proposed framework.

Keywords: Named Entity Recognition, Unsupervised Learning, Web Mining

1 Introduction

Entity detection is an important problem which has drawn much research efforts in the past decade. A lot of investigation has been done for detecting named entities from natural language texts or free texts such as [1, 2]. It can support a large number of applications such as improving the quality of question answering [3]. In this paper, we investigate the problem of detecting named entities from Web content associated with semi-structured or tabular text data records as shown in Fig. 1 and Fig. 2, without manually labeled data. Some existing methods on detection also make use of unlabeled data using weakly-supervised method such as [4] and semi-supervised method such as [5]. However, these existing methods cannot effectively handle the detection task from such kind of text data. Another limitation of these methods is that they still need some manually labeled data.

The first kind of Web content that we wish to handle is a list of semi-structured text data records called a *semi-structured record set* as exemplified

[*] The work is supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510).

in Fig. 1, which is taken from CICLing 2013 website. It is composed of a set of record information typically arranged as a list of records. Within a record, there are fields with possibly completely different formats. However, similar fields across records are formatted in a similar manner. Moreover, it is highly likely that named entities, if any, found in similar fields in different records belong to the same entity type. For example, the text field with a link under the photo from each record in Fig. 1 belongs to person names.

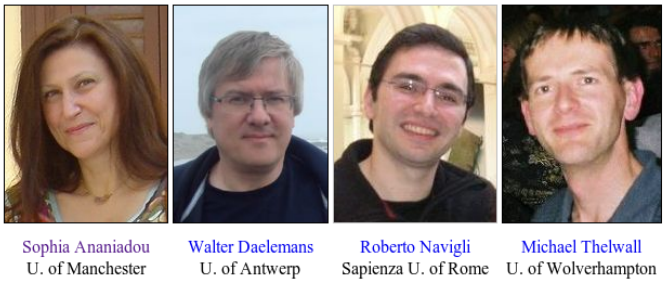
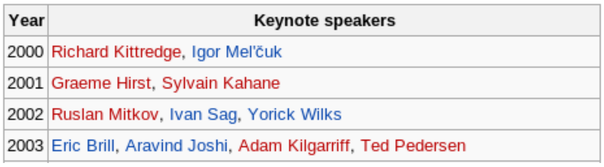


Fig. 1. An example of a semi-structured record set

The second kind of Web content is *tabular record set* as exemplified in Fig. 2. A tabular record set has a format similar to ordinary Web tables [6]. In general, multiple entities may exist in a single field. Most of fields under the same column share a common content type. A column may have a header text indicating the content of the column. For example, named entities found in the third column with header text "Keynote speakers" in Fig. 2 are person names.



Year	Keynote speakers
2000	Richard Kittredge, Igor Mel'čuk
2001	Graeme Hirst, Sylvain Kahane
2002	Ruslan Mitkov, Ivan Sag, Yorick Wilks
2003	Eric Brill, Aravind Joshi, Adam Kilgarriff, Ted Pedersen

Fig. 2. An example of a tabular record set

One common property for the above two content types is that they all have an inherent structure. For semi-structure record sets, each record can be segmented into fields. Corresponding fields with similar layout format in different records can be virtually aligned into a column. For tabular record sets, the structure can be readily obtained from HTML tags such as `<tr><td>`, with possible header text from `<th>` tags. The entities appeared in a particular column normally exhibit certain consistency between entities as well as header text, if any. This kind of structure information and possible column header text provide valuable guidance for the entity detection. We propose a framework that can exploit such underlying structure information via a transformation process facilitating collective detection. By incorporating existing named entity repositories such as DBpedia into the learning process via distant supervision, we do not require training labels on the data records. A collective detection model based on logical inference is proposed to consider the consistency among potential named entities as well as header text. Extensive experiments demonstrate the effectiveness of our framework.

2 Proposed Framework

2.1 Overview

Our framework focuses on two kinds of Web content mentioned above, namely, semi-structured record sets and tabular record sets. We transform these two kinds of record sets to a unified structure known as *structured field record lists*. A structured field record list consists of multiple records, with each record composed of multiple fields. A field is basically composed of text fragments possibly containing one or more, if any, named entities. Based on the layout format, corresponding fields in different records form a field column. A field column may optionally have a header text. We develop a component that is able to harvest semi-structured record sets from raw Web pages and transform the harvested record sets to structured field record lists based on the record field layout format. For tabular record sets, the detection and transformation are straightforward since we can directly examine HTML tags corresponding to tables.

The next component is to detect potential named entities from the generated structured field record lists. This component tackles the potential entity detection task for each record separately. To handle multiple entities possibly found in a field such as the records in Fig. 2, the detection is formulated as a sequence classification problem. Each record is tokenized as a token sequence and we aim to find the corresponding label sequence. We design labels based on the IOB format [7], and build a sequence classification model to predict the label for each token. To learn such a classification model, existing approaches rely on a large amount of training labels on the text data records. In contrast, our framework does not require training labels on the text data records. Instead, we leverage the existing large amount of labeled named entities from various external repositories such as DBpedia. We incorporate this external clue via distant supervision to guide the model learning. This paradigm is highly scalable in that it does not require tedious labeling effort.

After potential entities for each record are found as described above, the next component in our framework aims at taking advantage of the inherent structure information underlying the record list and considering the inter-relationships among records in the record list. One clue is that potential entities appeared in a particular field column of a record list generally share the same entity type. Another consideration is that some field columns may have header texts which can provide useful clues about the entity type of potential entities under those columns. A collective inference model is developed for incorporating all these clues based on logic paradigm. By exploiting such kind of structure information, better entity detection performance can be achieved.

2.2 Identifying and Transforming Semi-structured Record Sets

We first identify semi-structured record sets from Web page content. Then we conduct layout format driven alignment among the records in a record set resulting in the required structured field record lists.

Several methods may be applied to identify semi-structured record sets, such as MDR [8], DETPA [9], and RST [10]. MDR and DEPTA assume a fixed length of generalized nodes whereas RST relaxes this assumption by using a search structure called record segmentation tree which can dynamically generate subtree groups with different length. Moreover, RST provides a unified search based solution for region detection and record segmentation using a record segmentation tree structure. Our modified implementation of RST performs a top-down traversal detection in the DOM structure of a Web page.

After identifying semi-structured record sets, we make use of the partial tree alignment method [9] to conduct layout format driven alignment for the generation of structured field record lists. This approach aligns multiple tag trees of data records from the same record set by progressively growing a seed tree. The seed tree is chosen as the record tree with the largest number of data items because it is more likely for this tree to have a good alignment with data fields in other data records. Then the algorithm utilizes the seed tree as the core and aligns the remaining record trees with it one by one. We obtain the data fields from each record tree according to the alignment result and each record set is transformed into a structured field record list.

2.3 Potential Entity Detection with Distant Supervision

The aim of this component is to detect potential named entities for a particular record in a structured field record list. As mentioned above, we formulate it as a sequence classification problem, where each record is represented as a sequence of tokens and we aim at finding the label for each token. To achieve our goal, we make use of Conditional Random Field (CRF) [11] model. CRF is a discriminative undirected probabilistic graphical model, which enables us to include a large number of statistically correlated features. In particular we use linear-chain CRF, which considers conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ of input sequence $\mathbf{x}$ and label sequence $\mathbf{y}$ as depicted in Equation 1.

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\theta(\mathbf{x})} \exp(\sum_k \theta_k F_k(\mathbf{x},\mathbf{y})), \tag{1}$$

where $Z_\theta(x) = \sum_y \exp(\sum_k \theta_k F_k(\mathbf{x},\mathbf{y}))$ is the partition function and $F_k(\mathbf{x},\mathbf{y}) = \sum_i f_k(\mathbf{x}, y_i, y_{i-1}, i)$ is the feature function. The most probable label sequence for a given input sequence $\mathbf{x}$ is

$$\mathbf{y} = \arg\max_{\mathbf{y}} p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y}} \sum_{k} \theta_k F_k(\mathbf{y}, \mathbf{x}). \tag{2}$$

As mentioned in the overview, we do not require training labels on the text data records. Instead, we leverage the existing large amount of labeled named entities from the external repository DBpedia. However, this labeled entities cannot be directly used as training data for our classification model. Instead, we incorporate this external clue via distant supervision by making use of Generalized Expectation (GE) constraints. GE constraints were first proposed in [12] to incorporate prior knowledge about the label distribution into semi-supervised learning, and were later used in document classification [13], information extraction [12], etc.

The idea of GE constraints is to make use of conditional probability distributions of labels given a feature. For example, we may specify the probability that the token "George" labeled as PERSON should be larger than 80%. To capture this prior information, we introduce an auxiliary feature f as [[Entity Type=PERSON given Token="George"]]. The corresponding affine constraint is $E_{p_\theta}[f(x,y)] \geq 0.8$. Learning with GE constraints will attempt to match this kind of label probability distribution for a particular feature by model expectation on the unlabeled data. The GE constraints objective function term is in the form of $\triangle(\hat{f}, E_{p_\theta}[f(x,y)])$, where $\triangle$ is a distance function; $\hat{f}$ is the target expectation; and p_θ is the model distribution. For the CRF model, we set the functions to be conditional probability distribution and set the distance function as KL-divergence between two distributions. By adding the constraint term to the standard CRF log-likelihood function, we can incorporate such kind of external prior knowledge during the training process.

In our framework, we add features that a given test segment matches an existing entity name in DBpedia, in the form of `B-DBpedia-X` and `I-DBpedia-X`, where `X` is the entity type associated with DBpedia. We set the feature target distribution that most text segments with these features are labeled as the corresponding entity type. We may have different expectations for different entity types. For example, we have high confidence that text segments appeared in the DBpedia species should be the SPECIES type, since species names are quite limited and specialized. Another example is that we allow the text segment with `DBpedia-Work` feature to be detected as WORK type at a relatively low target distribution. This is due to the nature of WORK type that entities in this type have more varieties. For example, *Jane Eyre* may be classified as WORK if we are talking the novel, or be classified as PERSON if we are talking the woman with this name. By making use of GE constraints to guide the model training, we are able to incorporate distant supervision from external repositories.

In the process of feature extraction, we also include some commonly used features employed in linear-chain sequence CRF models. These features include factors between each token and its corresponding label, neighboring tokens and labels, transition factors between labels and some word pattern features. The learning process will capture the importance of each feature.

2.4 Collective Detection via Logical Inference

As mentioned in the overview of our framework, we aim to make use of the inherent structure information to consider the consistency among potential named entities as well as header text in a field column. We investigate a model using first-order logic to conduct logical inference and make decision on the predicted entity type. The first-order logic aims at modeling the knowledge about the decision process that resembles how human beings conduct logical inference. Another characteristic of the decision making model is that we wish to allow a principled handling of uncertainty in the decision making knowledge as well as the inference process. To achieve our goal, we employ the Markov Logic Network (MLN) model [14] in this component.

MLN model combines the Markov network with first-order logic, enabling uncertain inference. A MLN, denoted as L, consists of a set of formulas with weights (F_i, w_i), where F_i is a formula expressed in first-order logic. Together with a set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ with binary-valued node. Given different sets of constants C, we get different Markov networks sharing the same structure and parameters. The generated Markov network is called a *ground Markov network*. The probability distribution over possible worlds x specified by the ground Markov network is given by

$$P(X = x) = \frac{1}{Z}\exp(\sum_i w_i n_i(x)) = \frac{1}{Z}\prod_i \phi_i(x_i)^{n_i(x)} \tag{3}$$

where $n_i(x)$ is the number of true groundings of F_i in x. Given a ground Markov network, we can query the probability that whether a given ground atom is true. This inference procedure can be performed by MCMC over the minimal set of the ground network required to answer the query.

In our framework, we employ MLN to capture the following knowledge in the collective inference component:

- Potential named entities under the same field column tend to share the same entity type. This observation is derived from the inherent structure of record lists.
- If a given field column contains multiple potential entities, they likely share the same entity type. This is generally true due to the nature of the field such as the "Keynote speakers" column in Fig. 2.
- Potential named entities in the same field column should be consistent with the header text. For example, if header text is "Keynote speakers", the named entities under the column likely belong to the entity type Person.

Header text provides extremely useful clues for entity detection. To effectively make use of header information, we develop a method to incorporate header text with uncertainty handling by using the hypernym tree of an ontology such as WordNet [15]. In the beginning, we manually associate a set of ontology concepts for each entity type $c \in \mathcal{C}$, denoted as OC_c according to the intended meaning of the entity types for the application. For example, OC_{Work} contains the concepts "painting, picture (3876519)" and "album, record album (6591815)", where each concept is denoted by the synonym set with the concept ID in the parenthesis. Given an input header text in the form of noun phrase, we preprocess the header text with noun phrase chunker and identify the core term, denoted as ct. If the core term is in the plural form, its singular form is returned. For example, the term "speaker" in "Keynote speakers" is identified as the core term. Then we lookup the core term in the hypernym tree of WordNet to obtain the concepts that contain the core term, detored as OC_{ct}. Let $OC_{ct,c}$ denote the concepts in OC_{ct} that are in the hyponym paths of the concepts in OC_c. Let $\mathcal{C}' = \mathcal{C} \cup \{\text{Non-entity}\}$, and $OC_{ct,\text{Non-entity}}$ denote the concepts in OC_{ct} that are not in the hyponym paths of any concept in OC_c. The probability that the core term ct is associated with an entity type c is calculated as:

$$P(c|ct) = \frac{OC_{ct,c}}{\sum_{c' \in \mathcal{C}'} OC_{ct,c'}}. \tag{4}$$

To combine different clues, we define the predicates as shown in Table 1. The variable `entity` represents the detected potential named entities; `column` represents the field column; `type` represents the entity types. We design the following logical formulas, namely, from LF1 to LF4.

Table 1. List of MLN predicates

Predicate	Meaning
EntityInColumn(entity, column)	column information
ColumnHeaderSimilarToType(column, type)	header information
ColumnDominantType(column, type)	column dominant entity type
EntityInitialType(entity, type)	initial type given by detection phrase
EntityFinalType(entity, type)	final type after logical inference

$$\text{EntityInColumn(e,c)} \wedge \text{EntityInitialType(e,t)} \Rightarrow \text{ColumnDominantType(c,t)} \tag{LF1}$$

The formula LF1 expresses an observation corresponding to a field column. The more detected named entities from a single column that share the same entity type, the more likely that the field column contains that type of entities. A field column may contain multiple types of entities, each detected entity will contribute to the column global entity type. Note that the "+" symbol beside the variable t means that we will expand this formula with each possible groundings of t.

$$\text{ColumnHeaderSimilarToType(c,t)} \Rightarrow \text{ColumnDominantType(c,t)} \tag{LF2}$$

The formula LF2 incorporates the column header information for a given column. If the associate probability of the header text in the column c with an entity type t expressed in Equation (4) exceeds a threshold, then we add the corresponding positive evidence predicate ColumnHeaderSimilarToType(c,t). Note that header text may indicate multiple potential entity types. For example header text "Member" may contain list of organizations, or list of person names. Together with the formula LF1, we can infer the probability of global entity type for a field column.

$$\textsc{EntityInitialType}(\textsc{e,t}) \Rightarrow \textsc{EntityFinalType}(\textsc{e,t}) \qquad \text{(LF3)}$$

The formula LF3 indicates that the final entity type for a potential named entity e tend to be consistent with the original one. We observe that our sequence classification model can detect most of the named entities correctly, thus we give this formula a relatively high weight.

$$\textsc{EntityInColumn}(\textsc{e,c}) \wedge \textsc{ColumnDominantType}(\textsc{c,t}) \Rightarrow \textsc{EntityFinalType}(\textsc{e,t}) \qquad \text{(LF4)}$$

Besides the original type given during the detection phrase, the final entity type also depends on the column c where the entity e is located as shown in LF4. Field labels tend to be consistent with the column global entity type. The influence of column global entity type will increase as we have higher confidence on column entity type.

We can handle the situation that a column may have multiple global named entities. In this case, each field contains multiple named entities with different types.

3 Experiment

3.1 Experiment Setup

For the semi-structure record sets, we harvested from Web as described in Section 2.2. For the tabular record sets, we collected from a subset of the table corpus as mentioned in [16]. As a result, we collected 3,372 semi-structured and tabular record sets in total. Note that all these record sets do not have training labels. The number of records in a record set ranges from 2 to 296, with average 30. For the purpose of evaluation, we recruited annotators to find the ground truth named entities and provide labels on a subset of our full dataset. The number of record sets in this evaluation set is 650 composed of 16,755 true named entities.

We focused on the detection of five types of named entity: Organization, Person, Place, Work, Species. The meaning of these five types is exactly the same as in DBpedia. For example, Work includes artistic creations such as films, albums or songs. The remaining entity types are self-explanatory. We used DBpedia 3.8 published in August 2012 and indexed all the entity names using Apache Lucene for fast lookup when extracting CRF features.

We also implemented a comparison model known as *Repository Supervised Model*. This model checks each text segment against DBpedia and finds the corresponding entity type if exists. If a text segment corresponds to multiple named entities of different types in DBpedia, we randomly selected one.

Besides our full model, we also investigate a model known as *Our Model Without Collective Inference*. This model is essential our proposed model, but omitting the collective inference part. By comparing our proposed model with this one, we can investigate the benefit of the collective inference component.

We implemented the sequence classification model based on the open source MALLET [17] package, which provides implementation for linear-chain CRF with GE constraints. The collective logical inference is implemented based on the Alchemy[3] package, which provides functions for MLN inference. We manually assign weights to the formulas based on our prior knowledge. Specifically, we set w_1 as 1.0, w_2 as 5.0, w_3 as 2.0, and w_4 as 1.0. Our experiments show that the parameters are not sensitive to the final performance much.

3.2 Evaluation result

We use standard evaluation metrics, namely, precision P, recall R, and their harmonic mean F1 where $F1 = 2 * P * R/(P + R)$. We followed CoNLL-2003 evaluation procedure which only counts the exact match for entity names. Table 2 shows the performance of our experiment.

Table 2. Experimental result

Model	Measure	Organization	Person	Place	Species	Work	Overall
Repository Supervised Model	Precision	61.63%	78.33%	26.31%	93.05%	54.34%	60.44%
	Recall	50.06%	42.05%	11.10%	32.25%	44.55%	38.56%
	F1-score	55.24%	54.73%	15.62%	47.90%	48.96%	47.08%
Our Model Without Collective Inference	Precision	75.95%	64.77%	44.81%	89.43%	68.32%	66.31%
	Recall	70.60%	56.90%	17.21%	100.00%	48.63%	48.70%
	F1-score	73.18%	60.58%	24.86%	94.42%	56.81%	56.16%
Our Full Model	Precision	69.54%	72.63%	81.18%	100.00%	64.87%	70.46%
	Recall	83.17%	75.99%	44.64%	100.00%	86.40%	74.79%
	F1-score	75.74%	74.27%	57.60%	100.00%	86.40%	72.56%

From the evaluation result, it is clear that our proposed framework outperforms the Repository Supervised model significantly by over 20% relative F1 score improvement. The average recall for the Repository Supervised Model is only around 40%, meaning that more than half of the named entities in the evaluation set are not present in DBpedia. Our proposed framework successfully detects many previously unseen named entities with high precision.

Compared to the Repository Supervised model, our model without collective inference still improves the performance by about 10%. This result demonstrates

[3] Available at http://alchemy.cs.washington.edu

the effectiveness of the sequence classification model, which can capture large amount of features such as word capitalization, neighborhood labels, and boundary tokens across the record. Even though we do not use any labeled records as training data, the distant supervision with existing repository named entities still leads to good performance.

With the collective inference component, our full model further improves the performance. By taking advantage of the inherent structure of record set, we can discover more named entities with higher precision.

4 Related Work

Some methods have been proposed to detect entities from Web pages. For example, Limaye et al. developed a system that can find entities and relationships [16]. It mainly recognizes terms in the Web content that are some known entities found in a database, known as a catalog. The main characteristic of their method is to allow approximate matching between the terms in the Web text and the entity in the catalog. Kulkarni et al. proposed a method for matching spots on Web pages to Wikipedia entities [18]. However, all these methods dealing with Web texts assume that all potential entities detected are known entities. In contrast, our proposed framework is able to detect entities not already seen before.

Recently, researchers explore another valuable information resource, namely search log, to conduct entity extraction or attribute acquisition [19, 20, 21, 22]. In [19], a seed-based framework was proposed to allow weakly supervised extraction of named entities from Web search queries by calculating the similarity score between the search-signature vector of a candidate instance and the reference search-signature vector of a seed class. In [21], Guo et al. attempted to use a topic model to identify named entities in queries, and they showed that around 70% of the real search queries contain named entities. The methods in the above works are not applicable for the task we tackle in this paper due to data characteristics.

Currently, the state-of-the-art method for NER from free text is based on Conditional Random Fields [2, 23]. This approach is already applied in the entity detection flourishing short tweets under the combination with other models [24, 25]. However, these works are not suitable for our text content due to the nature of text data records. Moreover, we do not have manual labels on the text data records. In addition, the inter-dependency among the records in the same record set cannot be taken into account in traditional NER methods.

Distant supervision has been employed in various tasks such as relation extraction [26, 27], sentiment analysis [28, 29], and entity extraction from advertisements or tweets [30, 31]. As far as we know, our work is the first one that applies distant supervision on entity extraction from semi-structured data records using the generalized expectation model.

5 Conclusions and Future Work

We have proposed a new framework for detecting named entities from semi-structured web data including semi-structured and tabular record sets. We transform them into a unified representation, and then use a primarily unsupervised CRF model trained with GE constraints. We also propose a collective logical inference method that enables us to incorporate the underlying structure and header text information in record lists. We demonstrate the effectiveness of our framework through extensive experiments.

We intend to develop a more efficient training algorithm. Currently CRF training with GE constraints can only handle local features. Therefore we need to use MLN to incorporate global constraints. We will investigate an integrated way to handle such capability in a unified manner.

References

[1] McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL. (2003)

[2] Sarawagi, S., Cohen, W.W.: Semi-markov conditional random fields for information extraction. In: NIPS. (2004) 1185–1192

[3] McNamee, P., Snow, R., Schone, P., Mayfield, J.: Learning named entity hyponyms for question answering. In: Proc. of the Third International Joint Conference on Natural Language Processing. (2008) 799–804

[4] Pasca, M.: Weakly-supervised discovery of named entities using web search queries. In: Proc. of CIKM. (2007)

[5] Suzuki, J., Isozaki, H.: Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In: Proc. of ACL-08: HLT

[6] Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webtables: exploring the power of tables on the web. Proc. VLDB Endow. **1**(1) (August 2008) 538–549

[7] Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. CoRR (1995)

[8] Liu, B., Grossman, R., Zhai, Y.: Mining data records in web pages. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD (2003) 601–606

[9] Zhai, Y., Liu, B.: Structured data extraction from the web based on partial tree alignment. IEEE Trans. on Knowl. and Data Eng. **18**(12) (December 2006)

[10] Bing, L., Lam, W., Gu, Y.: Towards a unified solution: data record region detection and segmentation. In: Proceedings of the 20th ACM international conference on Information and knowledge management. CIKM '11 (2011)

[11] Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. (2001) 282–289

[12] Mann, G.S., McCallum, A.: Simple, robust, scalable semi-supervised learning via expectation regularization. In: Proceedings of the 24th international conference on Machine learning. ICML '07 (2007) 593–600

[13] Druck, G., Mann, G., McCallum, A.: Learning from labeled features using generalized expectation criteria. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. (2008)

[14] Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1-2) (February 2006) 107–136

[15] Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11) (November 1995) 39–41

[16] Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. **3**(1-2) (2010)

[17] McCallum, A.K.: MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu (2002)

[18] Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text. In: Proc. of the Int. Conf. on Knowledge Discovery and Data Mining. (2009) 457–465

[19] Paşca, M.: Weakly-supervised discovery of named entities using web search queries. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. CIKM '07 (2007) 683–690

[20] Paşca, M., Durme, B.V.: Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In: ACL. (2008) 19–27

[21] Guo, J., Xu, G., Cheng, X., Li, H.: Named entity recognition in query. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. SIGIR '09 (2009) 267–274

[22] Jain, A., Pennacchiotti, M.: Open entity extraction from web search query logs. In: Proceedings of the 23rd International Conference on Computational Linguistics. COLING '10 (2010) 510–518

[23] Krishnan, V., Manning, C.D.: An effective two-stage model for exploiting non-local dependencies in named entity recognition. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. ACL-44 (2006)

[24] Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies. HLT '11 (2011)

[25] Ritter, A., Clark, S., Etzioni, M., Etzioni, O.: Named Entity Recognition in Tweets: An Experimental Study. In: 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011)

[26] Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. ACL '09 (2009) 1003–1011

[27] Surdeanu, M., McClosky, D., Tibshirani, J., Bauer, J., Chang, A.X., Spitkovsky, V.I., Manning, C.D.: A simple distant supervision approach for the tac-kbp slot filling task. In: Proceedings of the TAC-KBP 2010 Workshop. (2010)

[28] Purver, M., Battersby, S.: Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th Conference of the EACL. (2012)

[29] Marchetti-Bowick, M., Chambers, N.: Learning for microblogs with distant supervision: Political forecasting with twitter. In: EACL. (2012) 603–612

[30] Singh, S., Hillard, D., Leggetter, C.: Minimally-supervised extraction of entities from text advertisements. In: Human Language Technologies: The 2010 Annual Conference of the NAACL. HLT '10 (2010) 73–81

[31] Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B.S.: Twiner: named entity recognition in targeted twitter stream. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. SIGIR '12 (2012) 721–730