Comput Math Organ Theory (2008) 14: 248–262
DOI 10.1007/s10588-008-9029-z

Conditional random fields for entity extraction and ontological text coding

Jana Diesner · Kathleen M. Carley

Published online: 20 June 2008

Abstract Previous research suggests that one field with a strong yet unsatisfied need for automatically extracting instances of various entity classes from texts is the analysis of socio-technical systems (Feldstein in Media in Transition MiT5, 2007; Hampe et al. in Netzwerkanalyse und Netzwerktheorie, 2007; Weil et al. in Proceedings of the 2006 Command and Control Research and Technology Symposium, 2006; Diesner and Carley in XXV Sunbelt Social Network Conference, 2005). Traditional as well as non-traditional and customized sets of entity classes and the relationships between them are often specified in ontologies or taxonomies. We present a Conditional Random Fields (CRF)-based approach to distilling a set of entities that are defined in an ontology originating from organization science. CRF, a supervised sequential machine learning technique, facilitates the derivation of relational data from corpora by locating and classifying instances of various entity classes. The classified entities can be used as nodes for the construction of socio-technical networks. We find the outcome sufficiently accurate (82.7 percent accuracy of locating *and* classifying entities) for future application in the described problem domain. We propose using the presented methodology as a crucial step in the process of advanced modeling and analysis of complex and dynamic networks.

Keywords Ontological Text Coding · Semantic networks · Entity Extraction · Supervised machine learning · Conditional models · Conditional Random Fields

J. Diesner (✉) · K.M. Carley
School of Computer Science, Institute for Software Research, Center for Computational Analysis of Social and Organizational Systems (CASOS), Carnegie Mellon University, 1327 Wean Hall, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
e-mail: janadiesner@gmx.net

1 Introduction

One key challenge in Information Extraction is distilling instances of certain types of information from unstructured, natural language text data (McCallum 2005). In the case of Named Entity Recognition (NER), this includes *people*, *organizations*, *locations*, and *other* Named Entities (NE) that are referred to by a name (Bikel et al. 1999). For different domains, text sets and research questions, different types of information can be of interest. Such alternative sets of relevant entity classes can be specified and organized in ontologies or taxonomies.[1]

Previous research has shown that one area with a strong yet unsatisfied need for the automated extraction of various instances of entity classes from texts is the analysis of socio-technical networks such as business corporations, Web2.0 communities, governmental organizations, or covert networks (Feldstein 2007; Hampe et al. 2007; Weil et al. 2006; Diesner and Carley 2005; Carley 2002). We refer to the instances of entity classes as entities, and to the process of retrieving entities from texts as ontological text coding. The methodology presented herein facilitates ontological text coding by automatically identifying and classifying entities in texts, where the entity classes do not need to match the traditional set of NE. The entities that are retrieved as a result of this process may then be used as nodes for the construction of socio-technical networks. Such networks are typically represented and stored as relational data, in which the nodes are the entities of interest, and edges are the relationships between the nodes. In the case of social networks, for instance, people are represented as nodes that are tied together via friendship relations or co-authorship ties. We envision researchers and analysts in business and management, organization science and behavior, public policy, and linguistics and rhetoric, among others, applying the presented technique as one crucial step in the process of efficiently distilling relational data from text data sets.

2 Background

For text analysis projects with a focus on socio-technical systems, one applicable ontology is the meta-matrix (Krackhardt and Carley 1998; Carley 2002). The meta-matrix is a multi-mode, multi-plex model that describes the entity classes *agent, event, knowledge, location, organization, resource*, and *task*. Each entity can furthermore have attributes, e.g. the attribute of agent *John* might be *age, 42* and *gender, male*. In the meta-matrix, *time*, including both explicit dates such as *14-08-2008* and expressions such as *tomorrow morning*, is also modeled as an attribute.

For this project, we use the meta-matrix as an ontology with the mentioned entity classes organized on the same hierarchical level (see Fig. 1). The relationships among the entities within and across any entity class form certain types of networks. For example, a social network is composed of ties among agents, and a membership network consists of connections between agents and organizations. The meta-matrix model

[1] Ontology (Greek) is the study of being or existence. Taxonomy (Greek) is the practice and science of classification. We use both terms interchangeably.

Meta-Matrix	Agent	Event/Task	Knowledge	Location	Organization	Resource
Agent	Social Network (NW)	Assignment NW	Knowledge NW	Agent-Loc NW	Membership NW	Capabilities NW
Event/Task		Precedence NW	Knowledge Reqmt NW	Event-Loc NW	Org Assignmt NW	Resource Reqmt. NW
Knowledge			Information NW	Knowledge-Loc NW	Org Knowledge NW	Training NW
Location				Proximity NW	Org Loc NW	Resource Loc NW
Organization					Interorg. NW	Org. Capabilities NW
Resource						Resource NW

Fig. 1 Meta-matrix model: Types of nodes and relations

allows for analyzing socio-technical systems as a whole or in terms of one (one-mode network) or more (multi-mode network) of the cells contained in the model. This framework has been used to empirically assess organizational structure, dynamics, power, and vulnerability in a diversity of contexts such as situational awareness in distributed work teams (Weil et al. 2006), email communication in corporations (Carley et al. 2006), political debates and decision making processes (Hampe et al. 2007), brand communities (Feldstein 2007), and counter terrorism (Diesner and Carley 2005, 2006).

"Named Entity Recognition" typically refers to the extraction of named examples or instances of the entity classes *agent*, *organization* and *location*. In this paper we are concerned with developing a methodology and computational solution for the more general task of identifying both named and unnamed examples of entities from an arbitrary ontology or set of entity classes. For example, we might be interested in tasks (e.g. *signing a contract*), resources (e.g. *vehicles*) or knowledge (e.g. *expertise in data analysis*). We refer to this task with the broader term "Entity Extraction".

The following example illustrates the Entity Extraction task. In the following excerpt from a UN News Service (New York) article released on 12-28-2004 we have *underlined* the entities relevant with respect to the meta-matrix categories.

> *Jan Pronk*, the Special Representative of Secretary-General *Kofi Annan* to *Sudan*, today called for the immediate return of the *vehicles* to *World Food Programme (WFP)* and *NGOs*.

From this text snippet, the network shown in Fig. 2 can be extracted. Please note that in this paper we focus on extracting and classifying nodes, while disregarding how they are linked into statements. For simplicity, the link formation approach taken for this example is based on word proximity in the text.

We define Entity Extraction as a two step process. In the identification step, terms that can be associated with an entity class of the ontology under consideration need to be correctly located in the texts. For this paper, the meta-matrix serves as the ontology. As terms we consider unigrams (e.g. *WFP*) as well as meaningful N-grams (e.g. the trigram *World Food Programme*). Identification implies the correct location of term boundaries from their beginning to their end. In the subsequent classification step, the identified entities need to be classified as one or more of the applicable entity classes. Mapping text terms to entities classes is a non-exhaustive and non-exclusive

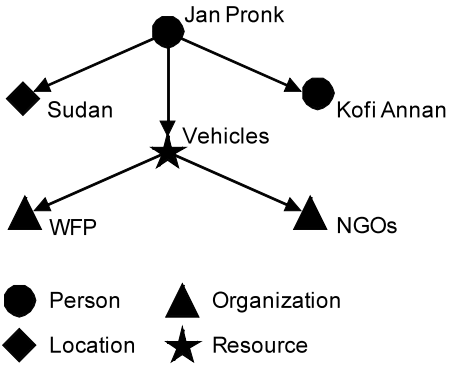

Fig. 2 Sample network. Nodes are identified from sample text and classified according to the meta-matrix model

process. "Non-exhaustive" means that not all terms in the texts need to be mapped to a class. Typically, words forming a large proportion of a text are irrelevant (e.g. *the*, *today*, *called*). In the sample text shown above for example, only 11 out of 28 words map to meta-matrix categories. In more randomly picked examples and across corpora, this ratio is likely to be much smaller. "Non-exclusive" means that relevant terms might be associated with one or more entity types, depending on the given context. For example, *World Food Programme* can be a resource in the context of providing aid, and an organization in the context of negotiating parties.

Ultimately, the goal of Entity Extraction in the described problem domain is the identification and classification of instances of various entity classes in text data as efficiently and accurately as possible. We expect the outcome of this process to facilitate the automated extraction of relevant nodes for coding texts as social-technical networks according to the meta-matrix model. Furthermore, we suggest exploring the methodology presented herein for its general applicability to ontological text coding based on a variety of ontologies.

If instances of the meta-matrix categories are to be identified in text data and subsequently classified, some list or mechanism needs to associate relevant words with one or more entity classes. Lists that contain the set of relevant terms for a given domain or research problem might exist in some cases, such as all agents in a parliament or all countries and languages in the world. However, such positive filters are unlikely to generalize well to unrelated projects, new data sets, or across time due to their incompleteness, static nature, spelling variations, and lack of synonym sets, among other issues. These limitations suggest that Entity Extraction is a non-deterministic process, which calls for an alternative solution.

If training data was available, one way to approach Entity Extraction could be supervised machine learning.

3 Data

Supervised machine learning requires labeled data for training and testing. More specifically, for Entity Extraction, a corpus is needed that is marked with the beginnings, endings, and classifications of relevant instances of entity classes. Traditional NER learning sets typically cover the categories person, organization, lo-

cation, miscellaneous (conglomerate of other named entities), and other (irrelevant words) (e.g. CoNLL 2003). While these entity classes can be mapped to parts of the meta-matrix (agent, organization, and location, respectively), other categories (e.g. task, resource, knowledge) are missing. Over the last decade, the classical set of NE has been extended to also cover time (e.g. dates), quantities (e.g. monetary values), geographical-political entities (e.g. countries), and facilities (e.g. buildings) (MUC 2006; LDC/ACE 2007), among others.

However, none of the existing NER corpora fully covers the entity classes of the meta-matrix. Our search for alternative, appropriately tagged data sets led us to the *BBN Pronoun Coreference and Entity Type Corpus* (BBN in the following) (Weischedel and Brunstein 2005). BBN was originally prepared for question answering tasks. The corpus contains approximately 1.1 million words organized in 95 XML files. BBN's categories map closely to the meta-matrix: all meta-matrix categories are represented, though mostly with a different name, while some additional classes are present in BBN that are irrelevant for the meta-matrix (e.g. sport games). In order to align BBN's categories with those of the meta-matrix, we matched and merged the BBN corpus' 12 NE types and 64 NE subtypes to fit the meta-matrix model. Table 1 provides details on the mapping process. In total, BBN contains 169,084 instances of meta-matrix categories. Figure 5 (in the results section) shows how many instances of each of the meta-matrix categories are contained in BBN. The *other* category in Fig. 5 is a collection of terms that are tagged as relevant instances in BBN, but that are irrelevant with respect to the meta-matrix (e.g. sport games). Working with the data revealed that the original BBN data had XML consistency issues, which we corrected for.

4 Methods

In order to select an appropriate learning technique, the characteristics of the training data need to be considered: First, the data is sparse. This means that even though a plethora of text data is available, only a small portion of the data is entities of interest, while the vast majority is irrelevant. In the BBN corpus, for instance, about 15 percent of the words represent instances of meta-matrix categories. Data sparseness is one characteristic feature of NER (McCallum 2005), which needs to be taken into consideration during the stages of method selection and implementation. Second, the data is sequential. This is because language is delivered and interpreted in a set order, and the elements that constitute the sequence (pairs of words and class labels) are not drawn independently from a distribution, but exhibit significant sequential correlation. For example, the tokens *World, Food*, and *Programme* are not independent from each other given the meaning of the trigram. In order to not only adequately represent the sequential nature of the data, but to also exploit this characteristic, a sequential learning technique seems appropriate.

4.1 Sequential learning for Entity Extraction

Sequential supervised machine learning techniques facilitate the modeling of relationships between nearby pairs of data points x and respective class labels y (Dietterich 2002). In our case, the data points x are text terms, and the class labels y are the

Table 1 Mapping of BBN categories to meta-matrix categories

BBN category	Meta-matrix category
Person Descriptor	agent
Person Name	agent
NORP	attribute
Percent	attribute
Quantity	attribute
Ordinal	attribute
Cardinal	attribute
Events Name	event
Disease Name or Descriptor	event
Law Name	knowledge
Language Name	knowledge
Facility Descriptor	location
GPE Descriptor	location
Facility Name	location
GPE Name	location
Location Name	location
Organization Descriptor	organization
Organization Name	organization
Product Descriptor	resource
Product Name	resource
Money	resource
Substance Name or Descriptor	resource
Date	time
Time	time
Plant Name or Descriptor	other
Animal Name or Descriptor	other
Work of Art Name	other
Contact info	other
Game Name or Descriptor	other

meta-matrix categories. Empiric work suggests that sequential, token-based models achieve higher accuracy rates for NER than more traditional models, such as Sliding Window techniques (Freitag 1997). Our goal with a sequential learning approach is to learn and construct a model h that for each sequence of (x, y) predicts an entity sequence $y = h(x)$ that generalizes with high accuracy to new and unseen text data. To illustrate this concept, the desired entity sequence y for our previously introduced sample sentence would be:

<agent begin, agent inside> , other other other other other-other <agent begin, agent inside> other <location>, other other other other other other other <resource> other <organization being, organization inside, organization inside> <organization> other <organization>.

Original sentence for reference:

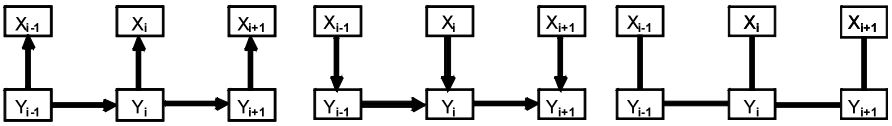

Fig. 3 Graphical structure of sequential models: First-order Hidden Markov model (HMM) (*left*), Maximum Entropy Markov Model (MEMM) (*middle*), Conditional Random Filed (CRF) (*right*)

> *Jan Pronk*, the Special Representative of Secretary-General *Kofi Annan* to *Sudan*, today called for the immediate return of the *vehicles* to *World Food Programme* (WFP) and *NGOs*.

Various models for working towards this goal exist. On a general level, these models can be divided into generative versus conditional (also known as discriminative) models. Figure 3 illustrates the models discussed for their applicability to Entity Extraction in the following. In this figure, the x's represent the words in a text, and the y's represent the respective feature that one wants to decode—in our case, whether a word is an instance of a meta-matrix class or not, and if so, a class label for that word. The directed graphs or models represent a distribution factored into a set of distributions where each node is conditioned on its parents. The undirected model represents a distribution factored into a set of "local likelihood" functions for each variable clique.

Generative Models estimate a joint distribution of the form $P(x, y, \ldots)$. Bikel et al. (1999) used a Hidden Markov Model (HMM), a special instance of generative models that has been successfully applied to Speech Recognition and other NLP tasks, for NER. Specifically, they deployed a HMM to decode the hidden sequence of NE that most probably has generated the observed sentences. Their implementation, named IdentiFinder, considers multiple words' features and achieves a NER accuracy of up to 94.9 percent. While NER accuracy rates gained with HMM are competitive with those achieved by using conditional models as will be shown later in this section, HMM lack the capability of directly passing information between separated y values. This information, which can be particularly valuable in the face of sparse data, can only be communicated indirectly through the y's that are intervening a separated pair of y's (Dietterich 2002). In the trigram *World Food Programme*, for instance, information about the class labels for *World* and for *Programme* cannot be communicated directly, but need to be channeled through the class label for *Food*. Another drawback of the HMM approach is that each x is generated only from the corresponding y, while information about nearby class labels cannot be exploited. This is another disadvantage exacerbated when working with sparse data.

An alternative to generative models are conditional models, which directly estimate a conditional distribution of the form $P(y|x)$. In other words, conditional models aim to find the most likely sequence of class labels y given an observed sequence of x, such as a sentence, without bothering to explain how the observed sequence was probabilistically generated from the y values—which in fact is irrelevant for the task at hand anyway. The main advantage of conditional models over generative ones is that they facilitate the usage of arbitrary features of the x's, such as global and long-distance features (Dietterich 2002). As a result, information about distant

class labels can be communicated directly in the model. For NER, a specific discriminative model, namely Conditional Random Fields (CRF) (Lafferty et al. 2001; Sha and Pereira 2003) has been shown to outperform generative models (Lafferty et al. 2001). For example, Lafferty et al. (2001) report an error rate of 5.69% for HMM, 6.37% for Maximum Entropy Markov Models (MEMM, another discriminative model, Borthwick et al. 1998), and 5.55% for CRF.

In comparative empiric studies on sequential learning models (such as the one cited in the previous paragraph), MEMM have led to higher error rate than generative models. Given that MEMM (as well as CRF) allow for using a bag of features f that depend on y_i and any property of sequence x, this drop in accuracy seems counterintuitive. It has been attributed to the label bias problem, which only MEMM exhibit. Why is that? MEMM is a log-linear model that learns the conditional probability $P(y_i|y_{i-1}, x_i)$. The learner uses maximum entropy to find the highest conditional likelihood of all x: $\prod P(y_i|x_i)$. Now label bias problems occur because all of the probability mass present in a class label y_{i-1} must be passed to the subsequent label y_i, even if the observed token x_i fits it only poorly or not at all (Lafferty et al. 2001). In CRF, this decision can be further delayed, until a better fit is detected.

4.2 Conditional Random Fields for Entity Extraction

Based on the properties of the described learning models as well as on the empiric results cited in the previous section, we decided to use CRF for Entity Extraction. In contrast to HMM and MEMM, CRF allow for modeling the relationship among y_i and y_{i-1} as a Markov Random Field (MRF) that is conditioned only on x. MRF are a general framework for representing undirected, graphical models. In CRF, the conditional distribution of an entity sequence y given an observation sequence (string of text data) x is computed as the normalized product of potential functions M_i (Lafferty et al. 2001; Sha and Pereira 2003):

$$M_i(y_{i-1}, y|x) = \exp\left(\sum_{\alpha} \lambda_\alpha f_\alpha(y_{i-1}, y_i, x) + \sum_{\beta} \mu_\beta g_\beta(y_i, x)\right) \quad (1)$$

In (1), the $f_\alpha(y_{i-1}, y_{i,}, x)$ component represents the transition feature function of an entire (that is, arbitrarily long) observation sequence as well as the entities at the current and preceding positions. The $g_\beta(y_i, x)$ component represents the emission feature function of an entity sequence from a term sequence. The feature vectors f_α and g_β are given, fixed, boolean feature vectors that depend on y_i and any property sequence of x. Note that f_α is an edge feature, while g_β is a vertex feature. Most of these features will be switched off or zero most of the time, and will be turned on only rarely. The word identity feature, which our implementation includes, for instance, is only positive when x contains that particular term. In our case, for each feature, the edge weights λ_α and the node weights μ_β are learned from the training data.

The potential functions are furthermore multiplied by $1/Z(x)$, where Z is a normalizing constant parameterized on data sequence x. As a result, the un-normalized scores of the potentials M_i are being normalized. Subsequently, the actual conditional

probability of the label sequence $P(y|x)$, where both x and y are both arbitrarily long vectors, is computed as:

$$p_\theta(y|x) = \frac{\prod_{t=1}^{n+1} M_i(y_{i-1}, y_i|x)}{\prod_{i=1}^{n+1} M_i(x)_{start,stop}} \quad (2)$$

For calculating the conditional probability, the length of the entire label sequence from its start at y_0 to its end plus one at y_{n+1} is considered. Overall, CRF enable the consideration of arbitrarily large numbers of features as well as long-distance information on at least x. In our experiments, for example, between 61,000 and 64,000 binary features were detected and used—far more than the typical handful of predetermined features for e.g. HMM. As a result, more information is exploited than with generative models. We argue that for sparse data, this exhaustive usage of available information is crucial.

As a starting point for implementing CRF we used a package provided by Sarawagi (n.d.). This framework provides a basic implementation of a CRF that can be adjusted and customized for specific types of CRF applications.[2] Features considered include word identity, transitions among class labels, starting features, ending features, word score features (the log of the ratio of current word with the label y to the total words with label y), and features for handling words that are new or have only been observed in other states so far. For training and testing, we included the *other* category (collection of all instances of those categories that are considered in BBN but not in the meta-matrix model) in order to have less sparse data.

Analogous to our definition of the Entity Extraction process, our CRF implementation consists of two steps: First, the CRF identifies relevant terms. These terms are marked as being a part of a relevant entity. If consecutive words are identified as belonging to one entity (e.g. World Food Programme), they are deterministically designated one concept. Second, the CRF is used to classify the identified relevant entities. In order to analyze and evaluate the accuracy achieved by both steps, we measured and report accuracy rates for each step separately.

5 Results

The overall accuracy of Entity Extraction stems from two components: the correct identification of entity boundaries (start to end), and the correct assignment of class labels to relevant terms. For validating our Entity Extraction implementation, a term had to be completely correctly located as well as correctly classified in order to be counted as a correctly extracted entity. In our case, classification is a nine-fold decision: a relevant term can be an agent, event, knowledge, location, organization, resource, time, attribute, or other. Under the category *other* we collected those entity classes that considered in the BBN data, but irrelevant to the meta-matrix. Furthermore, we decided to treat the attribute *time* as a separate category, because users or

[2]The specific network that we implemented in CRF is the naïve model graph type, since this structure and characteristic correspond to the linear nature of text data.

analysts often need terms representing time to be clearly identified as a specific subset of information.

In order to assess the accuracy of our Entity Extraction system, we computed recall, precision, and the F-measure. These are standard measures for evaluating the performance of Natural Language Processing and Information Extraction techniques (Bikel et al. 1999). Recall measures what percentage of the entities contained in the test data has been correctly found and classified by our engine (see (3)). Thus, recall resembles coverage. Precision measures what percentage of the extracted entities, which may include false positives, has been correctly identified and classified (see (4)). Thus, precision represents accuracy.

$$\text{Recall} = \frac{\text{number of correctly identified and correctly classified entities retrieved}}{\text{number of correct entitites in test data}} \tag{3}$$

$$\text{Precision} = \frac{\text{number of correctly identified and correctly classified entities retrieved}}{\text{number of entitites retrieved}} \tag{4}$$

Typically, recall and precision are inversely related: one could for instance achieve a full score on recall with respect to finding entities by retrieving all words from the data and suggesting that they are relevant entities. In that case, however, the presumably high number of false positives would reduce the precision. The F-measures accounts for this tradeoff by computing the harmonic mean between precision and recall (see (5)).

$$F = \frac{\text{recal} \cdot \text{precision}}{0.5(\text{recal} + \text{precision})} \tag{5}$$

Supervised machine learning systems are typically validated by performing a k-fold cross-validation. We complied with this standard technique by running a ten-fold cross validation on the data: This procedure randomly picks 90 percent of the data and uses them for constructing a model h. The resulting model is then applied to the remaining ten percent of the data in order to determine how correctly h predicts the label sequences for this data fold. This process is repeated nine more times, and the final results (see Table 2) are averaged over all ten runs.

Our results suggest that overall our Entity Extraction system correctly locates and classifies about at least eight out of ten entities. Precision exceeds recall by 0.9 percent, and this difference is statistically significant.

Table 2 Results: Accuracy of entity extraction

	Recall	Precision	F-Value
Average	82.3%	83.2%	82.7%
Maximum	83.4%	84.5%	83.8%
Minimum	80.1%	81.7%	80.9%
Standard Deviation	1.0%	0.8%	0.9%

5.1 Error analysis

Figure 4 shows the ground truth and our results side by side: the left bar represents all instances of meta-matrix categories and the *other* category as tagged in BBN. The right bar represents the ratios of our accurate results and the different types of errors: overall, our system correctly identifies and classifies 82.7 percent (F-value, see also Table 2) of the instances of the categories contained in BBN. At the same time, our engine fails to correctly locate 6.4 percent of the entities contained in the data (false negatives with respect to identification), and misclassifies 12.1 percent of the correctly located entities (false negatives with respect to classification).

As Fig. 4 illustrates, the number of correctly found and classified entities (black section of right bar) plus the number of false negatives with respect to identification and classification (dark gray and light gray section of right bar, respectively) equals the number of all entities in BBN (entire left bar). On average, 5.4 percent of the entities suggested by our system are false positives (white section of right bar). False positives here are terms that our engine identifies and classifies as some meta-matrix category, although they are irrelevant terms according to the test data. A visual, qualitative inspection of the false positives that were returned by our system indicates that a fair amount of these entities could be considered as relevant hits with respect to the meta-matrix: the terms *king*, *specialist*, and *reader*, for example, were assigned to the agent class, and *Mississippi* and *Tokyo* were suggested to be locations.

The distribution of accuracy rates and error types for each category considered by our engine is shown in Fig. 6. In Fig. 6, 100 percent (y-axis) again represent accurate results plus false negatives, while false positives are placed above the 100 percent line. Relating those percentages to the total frequency of terms per category in the test data (Fig. 5) suggests that only for categories with very few training instances

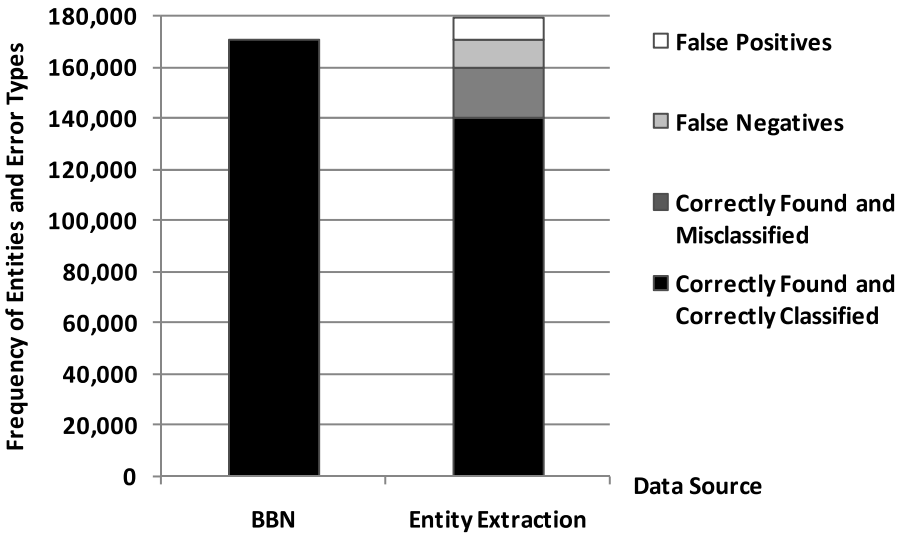

Fig. 4 Frequency of entities and error types per dataset

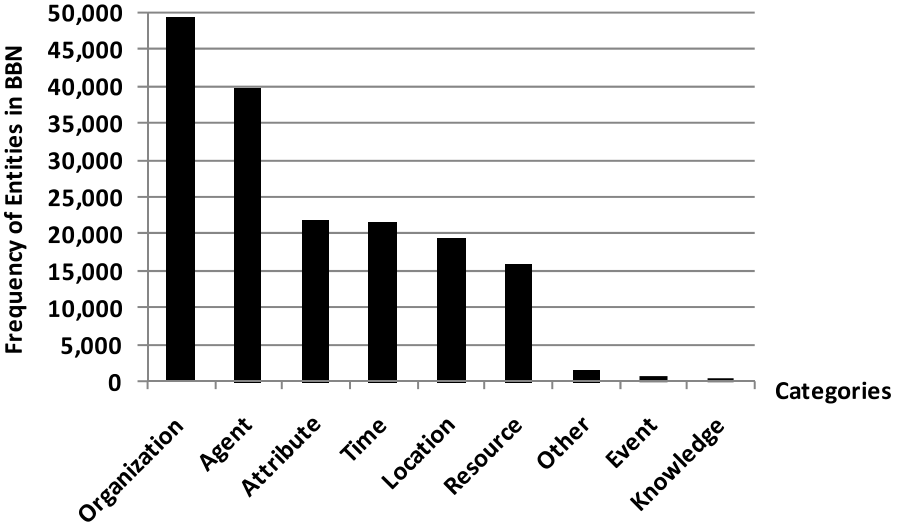

Fig. 5 Frequency of entities per category in BBN data

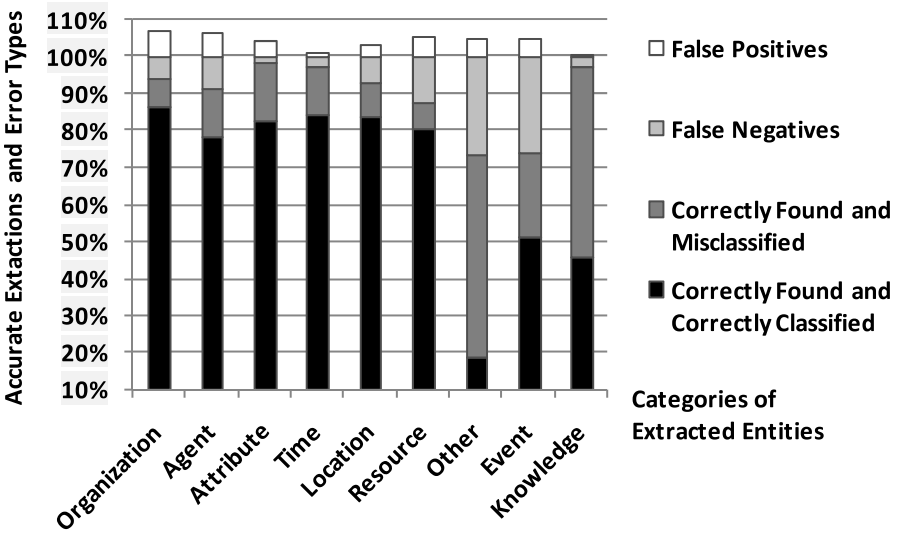

Fig. 6 Entity extraction: Ratios of correctly extracted entities and error types

(less than 1,500 for other, less than 1,000 for event and knowledge), unacceptably low accuracy rates are generated. Across the remaining categories (ranging from approximately 49,000 instances for organization to 16,000 for resource), we did not observe any relationship between larger amounts of training instances and higher accuracy rates per category, but fairly similar error rates. The *agent* class, for instance, is the second most frequent one in BBN, but shows the highest error rates among

the classes with 16,000 and more instances. Our engine performed worst by far on the *other* class—the only category that we do not consider in the meta-matrix or for further use.

Overall and to the best of our knowledge, no empiric point of comparison exists for our results. In comparison to classical NER (such as the studies cited in Sect. 4.1), our accuracy rates are considerably lower, which we partially attribute to three possible reasons:

First, we consider more categories (nine) than typical NER systems do (often four: people, places, organizations, other). Because of that, our classifier needs to pick one best category out of a larger pool of choices.

Second, we attempt to learn highly fuzzy categories such as knowledge, resource, and event. This might make term location and classification more difficult than in the case of classical NER. Why is that? The classical NE often exhibit certain properties or patterns (e.g. most names of people, places, and organizations are capitalized proper nouns in singular or plural), which the entities of our interest often do not show, and which probably are not compensated for by other properties. Furthermore, the entities considered in our study cover a much broader range of word identities—one of the features used by the learner—than classical NE.

Third, in Entity Extraction, it is even more likely than in NER that the same terms may be relevant in some sentences and irrelevant in others, depending on the context, domain, and rules for labeling the training data. For the learner, such terms are hard to distinguish from consistently relevant or irrelevant ones.

6 Summary and limitations

In this paper we have presented a new application of CRF and a new computational solution for mining texts for entities that are specified in non-traditional or user-defined ontologies. What could one expect from applying our engine in order to automatically finding and classifying potential nodes for the construction of socio-technical networks from texts? Out of 100 nodes contained in the actual data, 82 to 83 would be correctly found and assigned to a meta-matrix category, six to seven nodes would be missed; twelve nodes would be correctly found but misclassified, and five to six additional nodes would be suggested that may or may not be noise. Please note that these estimates rest on the theoretical assumption that our model generalizes to new data like it does to the test data that we used—an assumption that we have not tested so far, but plan to report on in the future. Overall, we assess the outcome in terms of accuracy rates and the resulting model as sufficiently successful for being applied in the described problem domain in the future. We envision Entity Extraction serving as a supplemental or alternative starting point for the process of automatically creating mappings from text words to entity classes.

Several limitations apply: First, CRFs enable us to detect relevant features along with their corresponding weights without having to have any preliminary or initial guess about what some of those features might be for a particular data set or domain. This means we can let the computer do all the work as long as we provide it with some labeled training data. However, such an uninformed global learning approach

comes at a price: Training the identifier and classifier on a reasonably sized data set and using a reasonable high iteration rate for the gradient can take a very long time. In our case, each of the ten runs that we performed for the ten-fold cross validation took about 20 hours to complete. This constraint can be alleviated to some degree by using more powerful hardware, especially such with more memory. However, this limitation made experimentation highly difficult and time consuming, which limited the practicality of exploring the parameter space and configuration, and tinkering with a variety of sample data types, sizes, and origins. Despite this constraint, we plan to perform further experiments with different parameter configurations of the CRF and other data sets. Second, in our current implementation and data set, relevant terms were associated with exactly one class label. The underlying ontology, however, allows for more flexible, non-exclusive label assignment. In the future we plan on modifying our system such that single terms can be mapped to multiple categories if applicable, and testing the resulting machinery with appropriate data. We understand that network models might need to be adjusted in order to represent this kind of ambiguity. Third, we argue that an ability to add, change, or remove labels from the used ontology is essential to having a flexible yet robust and sustainable learning and research process. While the meta-matrix currently has eight specific labels of interest, it is likely that the model may be altered as it evolves in the future. Finally, the limitations include a strong reliance on the training data for learning, which may or may not generalize well when this Entity Extraction program is run on different data sets. In the future, we will try to test and update our system based on alternative appropriately annotated test beds.

Acknowledgements An earlier version of this paper was presented at the North American Association for Computational Social and Organizational Science (NAACSOS) Conference, Atlanta, GA, June 2007, where it won the best student paper award. This research was supported in part by the ONR N00014-06-1-0104, ARI W91WAW07C0063, ARL DAAD 19-01-2-0009, and the NSF IGERT DGE-9972762 in CASOS. Additional support was provided by the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the Army Research Institute, the Army Research Lab, the National Science Foundation or the United States Government. We are grateful to Dr. William Cohen, CMU and Terrill Frantz, CMU, for discussing this project with us. We are also grateful to George Davis, CMU, for his comments on this paper.

References

Bikel DM, Schwartz R, Weischedel RM (1999) An algorithm that learns what's in a name. Mach Learn 34(1–3):211–231

Borthwick A, Sterling J, Agichtein E, Grishman R (1998) Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In: Sixth workshop on very large corpora association for computational linguistics, Montreal, QC, Canada, August 1998, pp 152–160

Carley KM (2002) Smart agents and organizations of the future. In: Lievrouw L, Livingstone S (eds) The handbook of new media. Sage, Thousand Oaks, pp 206–220

Carley KM, Frantz T, Diesner J (2006) Social and knowledge networks from large scale databases. In: 56th annual conference of the international communication association (ICA), Dresden, Germany, June 2006

CoNLL-2003 (2003) In: Proceedings of seventh conference on natural language learning (CoNLL-2003), Edmonton, Canada, May–June 2003

Diesner J, Carley KM (2005) Revealing and comparing the organizational structure of covert networks with network text analysis. In: XXV Sunbelt social network conference, Redondo Beach X, CA, February 2005

Diesner J, Carley KM (2006) Revealing social structure from texts: meta-matrix text analysis as a novel method for network text analysis. In: Narayanan VK, Armstrong DJ (eds) Causal mapping for information systems and technology research: approaches, advances, and illustrations. Idea Group, Harrisburg, pp 81–108

Dietterich TG (2002) Machine learning for sequential data: A review. In: Joint IAPR international workshops SSPR 2002 and SPR 2002, Windsor, ON, Canada, August 2002

Feldstein A (2007) Brand communities in a world of knowledge-based products and common property. Media in Transition MiT5, Cambridge, MA, April 2007

Freitag D (1997) Using grammatical inference to improve precision in information extraction. In: Fourteenth international conference on machine learning, workshop on automata induction, grammatical inference, and language acquisition, Nashville, TN

Hampe P, Hatzel I, Höhnsch J, Ueschner P (2007) Forschungsgruppe Transparentes Parlament, Ein neues Paradigma in den Sozialwissenschaften. Netzwerkanalyse und Netzwerktheorie, Frankfurt, Germany, September 2007

Krackhardt D, Carley KM (1998) A PCANS model of structure in organization. In: Proceedings of the 1998 international symposium on command and control, research and technology, Monterrey, CA, June 1998, pp 113–119

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Eighteenth international conference on machine learning (ICML-01), San Francisco, pp 282–289

Linguistic Data Consortium (LDC) Automatic Content Extraction (ACE) (2007) http://projectsldcupennedu/ace/ DARPA. Accessed March 20 2007

McCallum A (2005) Information extraction: distilling structured data from unstructured text. ACM Queue 3(9):48–57

Message Understanding Conferences (MUC) 6 (2006) Named entity task definition 1995. http://csnyuedu/cs/faculty/grishman/NEtask20book_1html. Accessed 11 March 2007

Sarawagi S (n.d.) CRF project page. http://crfsourceforgenet/. Accessed 20 March 2007

Sha F, Pereira F (2003) Shallow parsing with conditional random fields. In: Proceedings of human language technology, HLT-NAACL 2003, pp 213–220

Weil SA, Carley KM, Diesner J, Freeman J, Cooke NJ (2006) Measuring situational awareness through analysis of communications: A preliminary exercise. In: Proceedings of the 2006 command and control research and technology symposium, San Diego, CA

Weischedel R, Brunstein A (2005) In: BBN pronoun coreference and entity type corpus linguistic data consortium, Philadelphia, LDC2005T33

Jana Diesner is a doctoral student at Carnegie Mellon University (CMU), School of Computer Science (SCS), Center for Computational Analysis of Social and Organizational Systems, advised by Kathleen M. Carley. She has a M.A. in Communication from Dresden University of Technology and a M.S. in Computation, Organizations and Society from CMU, SCS. Her mission is to span the boundary between computational linguistics (aka natural language processing) and relational data analysis (aka network analysis). Her work is driven by her search for a better understanding of the co-evolution and interplay of the semantics and mechanics of real-world networks.

Kathleen M. Carley is a Professor of Computation, Organizations and Society at Carnegie Mellon University in the School of Computer Science and director of the center for Computational Analysis of Social and Organizational Systems (CASOS). She received her Ph.D. in Sociology from Harvard and has published multiple book and over 100 articles in this area. Her research combines cognitive science, social networks and computer science to address complex social and organizational problems.

Her specific research areas are dynamic network analysis, computational social and organization theory, adaptation and evolution, text mining, and the impact of telecommunication technologies and policy on communication, information diffusion, disease contagion and response within and among groups particularly in disaster or crisis situations.

She and the members of CASOS have developed infrastructure tools for analyzing large scale dynamic networks and various multi-agent simulation systems.