Conditional Random Fields for Entity Extraction and Ontological Text Coding

Jana Diesner
Carnegie Mellon University
jdiesner@andrew.cmu.edu

Kathleen M. Carley
Carnegie Mellon University
kathleen.carley@cs.cmu.edu

Abstract

Previous research has shown that one field with a strong yet unsatisfied need for automated extraction of instances of various entities classes from text data is the analysis of socio-technical systems (Carley, 2002; Diesner & Carley, 2005). Domain-specific entity classes and the relations between them are often specified in ontologies or taxonomies. We present a Conditional Random Field-based approach to distilling a non-canonical set of entities, which is defined in an ontology that originates from organization science. The supervised learning technique applied herein facilitates the derivation of relational data from corpora by locating and classifying instances of various entity classes. The classified entities can then be used as nodes for the construction of socio-technical networks. We envision researchers to use the presented methodology as one crucial step in the process of advanced modeling and analysis of complex and dynamic real-world organizations or networks. We find the outcome, particularly in the critical recall statistic, sufficiently successful for being applied in the described problem domain in the future.

Contact:
Jana Diesner
Carnegie Mellon University
School of Computer Science, Institute for Software Research
Center for Computational Analysis of Social and Organizational Systems (CASOS)
Computation, Organization, Society (COS) Program
Pittsburgh, PA 15213
Email: jdiesner@andrew.cmu.edu

Key Words: Ontological Text Coding, Semantic Networks, Entity Extraction, Machine Learning, Conditional Models, Conditional Random Fields

Acknowledgement: This research was supported by MURI: AFOSR, 600322, GRGMASON, ONR: ONR (IDA), N00014-06-1-0772, ARL: ARL, DAAD 19-01-2-0009, IGERT: NSF DGE-9972762. 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 external sponsoring agencies. We are grateful to Dr. William Cohen, CMU and Terrill Frantz, CMU, for discussing this project with us.

Conditional Random Fields for Entity Extraction and Ontological Text Coding

Jana Diesner and Kathleen M. Carley

1. Introduction

The 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), for instance, the relevant types of information are typically people, organizations, locations and other Named Entities (NE) that are referred to by a name (Bikel, Schwartz & Weischedel, 1999). Alternative sets of relevant classes or entities can be defined in ontologies or taxonomies.

Previous research has shown that one field with a strong yet unsatisfied need for the automated extraction of various entities is the analysis of socio-technical networks such as business corporations, governmental organizations or covert networks (Carley, 2002; Diesner & Carley, 2005). We envision researchers in the field of organizational science to apply entity extraction as one crucial step in the process of distilling relational data from text collections. By using the methodology presented herein, such relational data can be derived from corpora by locating and classifying instances of various entity classes, where the entity classes do not need to match the canonical set of NE, but might be specified in domain-specific ontologies. These classified entities can then be used as nodes for the construction of socio-technical networks.

2. Background

For corpora or text analysis projects with a focus on organizational science and behavior, one applicable ontology is the meta-matrix (Krackhardt & Carley, 1998; Carley, 2002). The meta-matrix is a multi-mode, multi-plex model that contains the following entity classes: agent, knowledge, resource, task, event, organization, location, time. Each instance of an entity class can furthermore have attributes, e.g. the attribute of agent *John* might be *age, 42* and *gender, male*. The relations among the elements within and across any entity classes form certain types of networks (see Figure 1). For example, a social network is composed of relations among agents, and a membership network consists of connections among agents and organizations. The meta-matrix model allows for analyzing socio-technical systems as a whole or in terms of one or more of the networks contained in the model. This ontological schema has been used to empirically assess power, vulnerability, and organizational change in a diversity of contexts such as situational awareness in distributed work teams, email communication in business corporations and counter terrorism (Carley, Frantz & Diesner, 2006; Diesner & Carley, 2005; Weil et al., 2005).

Meta-Matrix	**Agent**	**Knowledge**	**Resource**	**Event**	**Organization**	**Location**
Agent	Social nw	Knowledge nw	Capabilities nw	Assignment nw	Membership nw	Agent loc. nw
Knowledge		Information nw	Training nw	Knowledge requirement nw	Org. knowledge nw	Knowledge loc. nw
Resource			Resource nw	Resource require-ment nw	Org. Capabilities nw	Resource loc. nw
Events				Precedence nw	Org. Assignment nw	Task/Event loc. nw
Organization					Interorg. nw	Org. loc. nw
Location						Proximity nw

Figure 1: Meta-Matrix Model: Types of Nodes and Relations

We refer to the task of locating and classifying terms that represent instances of entity classes of the meta-matrix or of other models or ontologies that deviate from the classical set of NE in text data as Entity Extraction (EE). With this term we deviate from the term NER in order to adequately account for the fact that for the given task not only *named* entities are relevant, but also more fuzzy entities, such as tasks (e.g. signing a contract) and resources (e.g. vehicles), which are not necessarily referred to by a name. The following excerpt from an UN News Service (New York) article released on 12-28-2004I illustrates the EE task:

> 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.

Underlined are the entities relevant with respect to the meta-matrix. From this text snippet, the following network can be extracted (please note that here we focus on extracting and classifying relevant nodes, while disregarding how they are linked into statements):

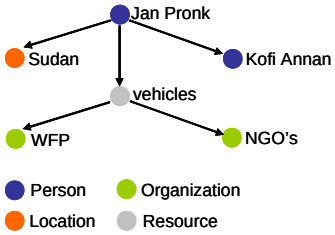

Figure 2: Sample network in which nodes are identified from sample text and classified according to the meta-matrix

We define EE as a two step process. First, terms that can be associated with an entity class of the ontology under consideration (in the case of this paper the meta-matrix model) need to be identified. As terms we consider unigrams (such as WFP) as well as meaningful N-grams (such as World Food Programme). Identification refers to correctly locating the boundaries (begin and end) of an entity in a text. Second, 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, non-exclusive process. Non-exhaustive means that not all text terms need to be mapped to a category. In fact, most terms are irrelevant (e.g. the, today, called, etc.). 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 might be a resource in the context of aid provision, and an organization in the context of negotiating parties.

Ultimately, the goal with EE for the described problem domain is the identification and classification of instances of certain entity classes in text data as accurately as possible. We expect the output 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.

3. Data

Supervised learning (explained in more detail under 4. Methods) requires tagged training and test data. More specifically, for EE, a corpus is needed in that the beginning and end of instances of entity classes are marked as such. Traditional NER text sets cover the entities person, organization, location, miscellaneous and other (e.g. CoNLL, 2003). While these categories can be mapped to parts of the meta-matrix (agent, organization, location), the entities knowledge, resource, task, event and time are missing. Over the last decade, the classical set of NE has been extended to also cover e.g. time (e.g. date), quantities (e.g. monetary values), geographical-political entities (e.g. countries), and facilities (e.g. buildings) (MUC 2006, LDC/ACE, 2007).

Unfortunately, none of the existing NER sets fully resembles the entity classes of the meta-matrix. In order to solve this problem, we searched for corpora tagged for other purposes. This search led us to the "BBN Pronoun Coreference and Entity Type Corpus", which was originally annotated for question answering tasks (BBN, 2005). The BBN corpus closely resembles the meta-matrix such that all meta-matrix entities are represented (mostly with a different name), while some additional entities are present in BBN that are irrelevant for the meta-matrix. The BBN corpus contains 1,133,218 words organized in 95 files. We matched and merged BBN's 12 NE types and 64 subtypes to the meta-matrix categories (the Appendix provides details on this mapping and matching). Though a valuable tagged data set, the original BBN data had XML consistency issues, which we corrected for. The final data set we worked with consisted of 95 tagged XML documents in that tags represent meta-matrix entity classes only.

4. Methods

If instances of the meta-matrix categories are to be identified in text data and classified, some list or mechanism needs to associate relevant words with one or more categories. Lists that contain the relevant terms for a given domain or research problem might exist in some cases (such all agents in a parliament, all countries and all languages in the world, etc.). However, such positive lists are unlikely to generalize well to unrelated projects or across time due to their incompleteness, static nature, and spelling variations, among other issues. This shows that EE is a non-deterministic process, which calls for an alternative solution.

Given the availability of training data, one way to approach this task is supervised machine learning. 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 only a small portion of the data is entities of interest, while the vast majority is irrelevant. For example, in the sample text shown on the previous page, only 11 out of 27 words match meta-matrix entity classes. In more randomly picked examples and across corpora, this ratio is likely to be even smaller. Data

sparseness is one characteristic feature of NER (McCallum, 2005), which needs to be addressed in the stages of model selection and implementation. Second, the data is sequential. This is because language naturally flows into one direction, and because the elements that constitute the sequence (pairs of data points and class labels) are not drawn iid (independent and identically distributed) from a joint distribution, but exhibit significant sequential correlation. For example, the tokens in the trigram *World Food Programme* are not independent from each other given the semantics of the trigram. In order to not to only adequately represent the sequential nature of the data, but to also exploit this characteristic, a supervised sequential learning technique seems appropriate.

4.1 Sequential Learning for Entity Extraction

Sequential learning facilitates the modeling of relationships between nearby pairs of data points *x* and respective class labels *y* (Dietterich, 2002). Recent 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 sequential learning is to construct a classifier *h* that for each sequence of (*x, y*), where *x* are the words in a sequence and *y* are the corresponding labels or meta-matrix entities, predicts with the highest accuracy possible an entity sequence $y = h(x)$ for unseen sequences of *x*. For the sample sentence on the previous page, e.g., the desired *y* would be

> agent, other other other other other-other agent other location, other other other other other other other other resource other organization and organization.

Various models exist for working towards this goal. Those models can be divided into generative versus conditional (aka discriminative) models. Figure 3 illustrates the models discussed for their applicability to EE in the following.

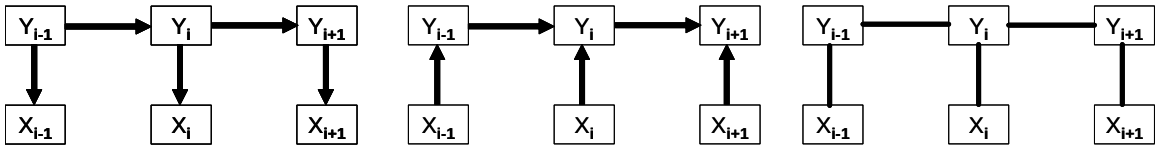

Figure 3: Graphical Structure of Sequential Models: HMM (left), MEMM (middle), CRF (right)

Generative Models estimate a joint distribution $P(x,y)$. 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 performed decoding via a HMM in order to find the sequence of hidden NE that most probably has generated an observed sentence. Their implementation, named IdentiFinder, considers multiple words features and achieves an 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 herein, HMM lack the capability of directly passing information between separated *y* values. This information, which can be particularly valuable in the face of spare data, can only be communicated indirectly through the *y's* that are intervening a separated pair of *y's* (Dietterich, 2002). Another drawback of HMM is that each *x* is generated only from the corresponding *y*, while *y's* surrounding the current *y* cannot be considered, which again might pose a serious disadvantage when working with sparse data.

An alternative to conditional models are discriminative models, which directly estimate $P(y|x)$. Thus, conditional models aim to find the most likely sequence of entities given an observed sequence, e.g. a sentence, without bothering to explain how the observed sequence was probabilistically generated from the *y* values. Therefore, 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). For NER, discriminative models such as Maximum Entropy Markov Models (MEMM) (Borthwick et al., 1998) and Conditional Random Fields (CRF) (Lafferty, McCallum, & Pereira, 2001; Sha & Pereira, 2003) have been shown to outperform generative models (HMM) (Lafferty, McCallum, & Pereira, 2001). For example, Lafferty et al. (2001) report an error rate of 5.69% for HMM, 6.37% for MEMM, and 5.55% for CRF.

In comparative empiric studies on generative model, MEMM have led to higher error rate than generative models (e.g. Lafferty, McCallum, & Pereira, 2001). 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 $P(y_i | y_{i-1}, x_i)$. The learner uses maximum entropy to maximize the conditional likelihood of all *x:* $\prod P(y_i | x_i)$. Now the label bias problems occurs because all of the probability mass present in y_{i-1} must be passed to y_i, even if x_i fits it only poorly or not at all (Lafferty, McCallum, & Pereira, 2001).

4.2 Conditional Random Fields for EE

Based on the empiric results presented by others, we decided to use CRF for the outlined EE task. 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 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, McCallum, & Pereira, 2001; Sha & Pereira, 2003):

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

Equation 1: Computation of Potentials

In equation 1, $f_\alpha(y_{i-1}, y_i, x)$ represents the transition feature function of an entire observation sequence as well as the entities at position i and the preceding position. The $g_\beta(y_i, x)$ component represents the emission feature function of a term sequence and an entity. F_α and g_β represent given, fixed boolean feature vectors that depend on y_i and any property sequence of x. Note that f_α is an edge feature, and g_β is a vertex feature. Most of these features will be 0 most of the time, and will be turned on only rarely (e.g. the word identity feature is only positive when x contains that particular term). For each feature, the weights λ_α or μ_β are learned from the training data. For CRF computation, equation 1 is multiplied by 1/Z(x), where Z is a normalizing constant over the data sequence x; so that un-normalized scores of the potentials M_i are being normalized. The conditional probability of the label sequence $P(y|x)$, where both x and y are arbitrarily long vectors, is computed as:

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

Equation 2: Computation of conditional probability of entity sequence *y*

In equation 2, $n+1$ is the length of the label sequence plus one, start = y_0 and end = y_{n+1}. Overall, CRF enable the consideration of arbitrarily large numbers of features as well as long-distance information on at least x. As a result, more information can be exploited than generative models can make use of, which is critical given data sparseness.

As a starting point for implementing CRF we used the CRF framework provided by Sunita Sarawagi of IIT Bombay (Sarawagi, n.d.). This framework provides a basic implementation of a CRF that can be adjusted and customized for specific types of CRF applications[1]. Features considered include the word identity, transitions among labels (including sequential information), start features, end features, word score features (log of the ratio of current word with the label y to the total words with label y), and features for dealing with yet unobserved words or words only observed in other states so far. Across multiple experiments, more than 10,000 binary features were detected. The EE process consists of two steps in our implementation. First, the CRF is used to locate the terms that are relevant entities. These terms are then marked as being a part of a relevant entity. Second, the CRF is used to classify the identified relevant entities. In order to do this, consecutive words that have been identified as belonging to entities are merged into one concept. This concept is represented as a concatenation of the consecutive entity words. In order to analyze and evaluate the accuracy achieved by both steps, we measure and report accuracy rates for both steps separately.

5. Results

The accuracy of EE has two components to it: the correct identification of entity boundaries (start and end), and the correct assignment of class labels. We evaluate the accuracy of our system in terms of recall, precision, and the F-measure (Bikel et al., 1999). Recall measures how many of the entities in the test data have been extracted. Thus, recall resembles coverage:

$$Recall = \frac{\#\ of\ correct\ entities\ identified\ by\ EE\ system}{total\ \#\ correct\ entities\ in\ test\ set}$$

[1] 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.

Precision measures how many of the extracted entities are actually correctly identified and classified. Thus, precision resembles accuracy:

$$Precision = \frac{\#\ of\ correct\ entities\ identified\ by\ EE\ system}{\#\ answers\ given\ by\ EE\ system}$$

Typically, recall and precision are inversely related. The F-measures accounts for this tradeoff; computing the harmonic mean between precision and recall:

$$F = \frac{recall * precision}{0.5(recall + precision)}$$

The validation effort for our EE implementation consists of two parts: 1) assessing the accuracy of locating entities and 2) assessing the accuracy of classifying the located entities (assigning a class label to them). For locating entities, we applied a two-fold cross validation, where 90% of the data were used for learning, and the remainder for validation. For classification, we set the number of iterations to 50, used 10 files for training and 45 files for validation. The entire procedure took about 3.5 hours to run. Please note that different validation strategies were applied for both steps due to practicality reason: The learning process for classification is computationally very expensive in terms space and time complexity of the CRF algorithm:

	Identification	Classification
Precision	75.51%	64.88%
Recall	52.33%	54.90%
F-Value	61.82%	59.47%

Table 1: EE results

For EE, to our knowledge, no empiric point of comparison for our results exists. In comparison to classical NER, our accuracy rates appear to be considerably lower, which we attribute to the learning of highly fuzzy categories such as knowledge, resource, event and tasks. We assume that for the learner, instances of those categories are hard to distinguish from irrelevant terms, e.g. because they do not exhibit a certain capitalization pattern, and because they cover a much broader range of word identities than classical NE. As a result, in EE, it is even more likely than in NER that some terms in some cases are relevant entities, while in other cases they are not, depending on the context, the domain, and the rules for labeling the training data. Overall, we assess the outcome, particularly in the critical recall statistic, sufficiently successful for being applied in the described problem domain in the future.

6. Limitations and Future Work

The global learning of features along with their corresponding weights comes at a price: Training the identifier and classifier while using a reasonable iteration rate for the gradient takes a very long time. This limitation can be addressed to some degree by using more powerful hardware, especially by using more memory. However, this limitation made experimentation highly difficult and time consuming, which can rather limit the practicality of exploring the parameter space and tinkering with a variety of sample data types, sizes, and origins. Furthermore, an ability to add, change, or remove labels from the used ontology is essential to having a flexible yet robust learning and research process. While the meta-matrix, today, 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 EE is run on unseen data.

References

Bikel, D., M., Schwartz, R., & Weischedel, R., M. (1999). *An Algorithm that Learns What's in a Name, Machine Learning* (Vol. 34, pp. 211-231): Kluwer Academic Publishers.

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). *Exploiting diverse knowledge sources via maximum entropy in named entity recognition.* Paper presented at the Sixth Workshop on Very Large Corpora. Association for Computational Linguistics, New Brunswick, New Jersey.

Carley, K. (2002), *Smart Agents and Organizations of the Future.* In L. Lievrouw and S. Livingstone, Eds., The Handbook of New Media Ch 12, Sage, Thousand Oaks, CA (2002), pp. 206-220.

Carley, K., Frantz, T., & Diesner, J. (2006). *Social and Knowledge Networks from Large Scale Databases*. 56th Annual Conference of the International Communication Association (ICA). Dresden, Germany, June 19-23, 2006.

CoNLL-2003. *Proceedings of Seventh Conference on Natural Language Learning* (CoNLL-2003), May/ June 2003, Edmonton, Canada. (2003).

Diesner, J., & Carley, K.M. (2005). *Revealing and Comparing the Organizational Structure of Covert Networks with Network Text Analysis.* XXV Sunbelt Social Network Conference, Redondo Beach, CA, February 16-20, 2005.

Dietterich, T. G. (2002). *Machine Learning for Sequential Data: A Review.* Paper presented at the Joint IAPR International Workshops SSPR 2002 and SPR 2002, August 6-9, 2002,, Windsor, Ontario, Canada.

Freitag, D. (1997). *Using grammatical inference to improve precision in information extraction.* In proceedings of the Fourteenth International Conference on Machine Learning, Workshop on Automata Induction, Grammatical Inference, and Language Acquisition, Nashville, TN, USA..

Krackhardt, D & Carley, K (1998). *A PCANS Model of Structure in Organization.* Proceedings of the 1998 International Symposium on Command and Control, Research and Technology, Monterrey, CA (June), pp. 113-119.

Lafferty, J., McCullum, A., & Perieira, F. (2001). *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* In proceedings of the Eighteenth International Conference on Machine Learning (ICML-01).

Linguistic Data Consortium (LDC). Automatic Content Extraction (ACE). http://projects.ldc.upenn.edu/ace/ DARPA. TIDES. URL: http://projects.ldc.upenn.edu/TIDES/. Retrieved March 18, 2007.

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

Message Understanding Conferences (MUC) 6, Named Entity Task Definition. 1995. URL: http://cs.nyu.edu/cs/faculty/grishman/NEtask20.book_1.html.

Sarawagi, S. (n.d.). CRF Project Page URL: http://crf.sourceforge.net/

Sha, F., & Pereira, F. (2003). *Shallow parsing with conditional random fields.* In Proceedings of HLT-NAACL, 213–220. Association for Computational Linguistics.

Weil, S.A., Carley, K.M., Diesner, J., Freeman, J., and Cooke, N.J. (2005). *Measuring Situational Awareness through Analysis of Communications: A Preliminary Exercise.* Submitted to the Command and Control Research and Technology Symposium 2006, San Diego, CA.

Weischedel, R. & Brunstein, A. (2005). *BBN Pronoun Coreference and Entity Type Corpus Linguistic Data Consortium*, Philadelphia. LDC2005T33.

Appendix

BBN	meta-matrix detailed	meta-matrix coarse
Person Descriptor	agent_general	agent
Person Name	agent_specific	agent
NORP	attribute	attribute
Events Name	event_specific	event
Disease Name or Descriptor	event_specific	event
Law Name	knowledge_specific	knowledge
Language Name	knowledge_specific	knowledge
Facility Descriptor	location_general	location
GPE Descriptor	location_general	location
Facility Name	location_specific	location
GPE Name	location_specific	location
Location Name	location_specific	location
Organization Descriptor	organization_general	organization
Organization Name	organization_specific	organization
Product Descriptor	resource_general	resource
Product Name	resource_specific	resource
Money	resource_specific	resource
Substance Name or Descriptor	resource_specific	resource
Date	time	time
Time	time	time
Percent	not applicable	not applicable
Quantity	not applicable	not applicable
Ordinal	not applicable	not applicable
Cardinal	not applicable	not applicable
Plant Name or Descriptor	not applicable	not applicable
Animal Name or Descriptor	not applicable	not applicable
Work of Art Name	not applicable	not applicable
Contact info	not applicable	not applicable
Game Name or Descriptor	not applicable	not applicable

Appendix: Mapping the BBN categories to the meta-matrix at different levels of detail