Development of NeuroElectroMagnetic Ontologies(NEMO): A Framework for Mining Brainwave Ontologies

Dejing Dou
Computer and Information Science
University of Oregon, USA
dou@cs.uoregon.edu

Gwen Frishkoff
Learning Research and Development Center
University of Pittsburgh, USA
gwenf@pitt.edu

Jiawei Rong
Computer and Information Science
University of Oregon, USA
jrong@cs.uoregon.edu

Robert Frank
NeuroInformatics Center
University of Oregon, USA
rmfrank@mac.com

Allen Malony
NeuroInformatics Center
University of Oregon, USA
malony@cs.uoregon.edu

Don Tucker
NeuroInformatics Center
University of Oregon, USA
Electrical Geodesics, Inc.
dtucker@egi.com

ABSTRACT

Event-related potentials (ERP) are brain electrophysiological patterns created by averaging electroencephalographic (EEG) data, time-locking to events of interest (e.g., stimulus or response onset). In this paper, we propose a generic framework for mining and developing domain ontologies and apply it to mine brainwave (ERP) ontologies. The concepts and relationships in ERP ontologies can be mined according to the following steps: pattern decomposition, extraction of summary metrics for concept candidates, hierarchical clustering of patterns for classes and class taxonomies, and clustering-based classification and association rules mining for relationships (axioms) of concepts. We have applied this process to several dense-array (128-channel) ERP datasets. Results suggest good correspondence between mined concepts and rules, on the one hand, and patterns and rules that were independently formulated by domain experts, on the other. Data mining results also suggest ways in which expert-defined rules might be refined to improve ontology representation and classification results. The next goal of our ERP ontology mining framework is to address some long-standing challenges in conducting large-scale comparison and integration of results across ERP paradigms and laboratories. In a more general context, this work illustrates the promise of an interdisciplinary research program, which combines data mining, neuroinformatics and ontology engineering to address real-world problems.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Data mining; J.3 [**Life and Medical Science**]: Neuroscience; I.2.4 [**Knowledge Representation Formalism and Methods**]: Ontology

General Terms

Theory, Design

Keywords

Ontology Mining, Clustering-based Classification, Temporal PCA, ERP, Semantic Web

1. INTRODUCTION

Research in cognitive and clinical neuroscience has given rise to a wealth of data over the past several decades. It is becoming increasingly clear that management and distribution of these data will require advanced tools for data representation, mining, and integration. In this paper, we propose a generic framework for mining and developing domain ontologies and apply it to mine brainwave (ERP) ontologies. Techniques are applied to several dense-array (128-channel) datasets acquired during studies of visual word comprehension. Development of these ERP ontologies will support future work on semantic mapping discovering, multi-modal data integration, and cross-laboratory data sharing.

1.1 EEG and ERP Data

Electroencephalography (EEG) is a widespread, noninvasive method for imaging brain activity. EEG data are acquired by placing sensors on the head to measure electrical signals that are generated in the cortex and conducted to the scalp surface. Compared with other noninvasive imaging techniques, such as Positron Emission Tomography (PET) and functional Magnetic Resonance Imaging (fMRI), EEG methods have two advantages: first, they provide a direct measure of neuronal activity (PET and fMRI measure the hemodynamic response, which is closely linked with neuronal activity), and second, they have excellent temporal resolution - on the order of milliseconds, compared with 6 seconds or more for hemodynamic measures. Given that most sensory-motor and cognitive processing takes place within a few hundred milliseconds, fine-grained representation of the time course of brain activity is extremely important. In addition, with the advent of dense-array methodologies, modern EEG methods are now characterized by high spatial (scalp topographic), as well as high temporal, dimensionality. With the application of tools for anatomical source localization, dense-array EEG can be used for non-invasive brain functional mapping, supporting a wide range of clinical and basic research applications.

Event-related potentials (ERPs) are derived by averaging across segments of EEG data, time-locking to events of interest (e.g., onset of a visual or auditory stimulus). Signals that are not event-related tend towards zero as the number of averaged trials increase. In this way, averaging increases the signal-to-noise ratio (SNR) and provides measures of electrical activity that are specifically linked to stimulus processing (e.g., Figure 1(A)).

At each time point, many parts of the brain may be simultaneously active, contributing overlapping (or "superposed") patterns to the measured signal. ERP research aims to separate and classify these patterns (or "components") and to relate them to specific brain and cognitive functions. Distinct patterns are characterized by their time course (e.g., early or late), polarity (positive or negative), and scalp distribution, or topography. For example, as illustrated in Figure 1, the "P100 component," which was extracted from the superposed data (A) using Principal Components Analysis [14] has a peak latency of approximately 100ms (B) and is positive over occipital areas of the scalp (C).

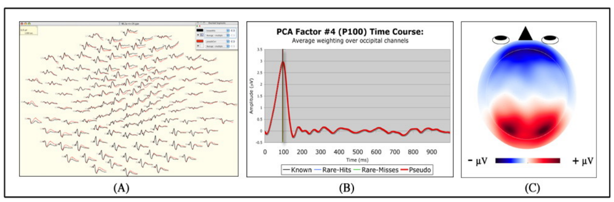

Figure 1: (A)128-channel EEG waveplot; positive voltage plotted up. (B) Time course of P100 factor for same dataset, extracted using Principal Components Analysis. (C) Topography of P100 factor (negative on top and positive at bottom).

Although there is general agreement on how to characterize ERP patterns or "components," in reality, such patterns can be difficult to identify, and definitions vary across research labs. Furthermore, methods for ERP data summary and analysis differ widely across research sites. This variability can make it hard to compare results across experiments and across laboratories, limiting the generalizability of research results and, therefore, the ability to generate high-level integration and interpretation of patterns.

1.2 Domain Ontologies and Ontology Mining

To address these issues, we have proposed a new framework, called "Neural ElectroMagnetic Ontologies," or NEMO. The NEMO project proposes to develop ontologies to support ERP data representation and integration.

In general, an ontology can be defined as the formal specification of a vocabulary of concepts and the relationships among them in a specific domain. In traditional knowledge engineering and in emerging Semantic Web research, ontologies play an important role in defining the semantics of data. The adoption of domain ontologies in biomedical research has enabled several major scientific advances [22], which are exemplified in projects such as the Gene ontology [19], UMLS [24] and the National Center of Biomedical Ontology [8]. Most biomedical ontologies are developed through a top-down or knowledge driven approach, i.e., domain experts define the concepts and relationships based on their domain knowledge with the help of ontology engineers. Currently there are no formal ERP ontologies, and in fact there is little neuroinformatics research in this important area, although there are a variety of statistical techniques that are emerging for analysis of spatiotemporal patterns in EEG and ERP research [17]. The reason for this gap may be linked to the absence of robust methods for identification of ERP patterns (concepts). Perhaps the greatest challenge, at this stage of the ontology development, is to develop and test a framework for separating and classifying complex spatiotemporal patterns that are superposed in measured EEG. In this paper, we describe some general concepts (i.e., patterns) and rules (i.e., the high-level pattern representations) that have emerged from our prior work on neuroscience, and detail the methodology used to mine and develop initial ERP ontologies which formally represent and store these concepts, their taxonomies and high-level representations (i.e., rules).

There are several popular ontology languages which are based on different logics, such as the Web Ontology Language (OWL [6]) based on description logic, KIF [20] based on first order logic and OKBC [5] (i.e., the protocol used by Protege-frames) based on frame-logic. In general, the vocabulary and relationships in an ontology can be roughly divided into several categories:

- *Class and Class taxonomy*: The basic structure of an ontology is a set of classes (types) arranged in a subclass hierarchy. Each class corresponds to a specific set of entities. The class names (terms) and hierarchy (taxonomy) are assumed to refer in well-defined ways to concepts and provide the basic metadata for various domains.

- *Relationships among classes and data types*: To represent the relationships between two classes, OWL uses ObjectProperties, OKBC uses slots and KIF uses binary predicates. The relationships can also be between one class and one data type (e.g., string and number). In that case, OWL uses DatatypeProperties, KIF and OKBC still use binary predicates or slots. If the relationships are n-ary (n>2) relations among different classes and data types. OWL divides the n-ary relations into multiple ObjectProperties and DatatypeProperties. KIF uses n-ary predicates and OKBC uses facets.

- *Arbitrary relationships or constraints among concepts*: The relationships can be among different properties (slots, facets), or between classes and properties. The constraints can be cardinalities or other more complex forms which are related to classes or properties. In general, the arbitrary relationships or constraints can be presented as axioms (i.e., logic rules) in ontology languages, and they can support reasoning about the concepts.

The objective of ontology mining is to mine domain specific (i.e., real world) data to acquire a vocabulary of concepts, to establish a concept taxonomy, and to discover the relationships among the concepts. In this paper, we will mainly use OWL to describe the classes, properties and axioms to be mined, since OWL has become the W3C standard web ontology language. We note that most concepts described in OWL can also be represented in KIF or OKBC.

The reminder of the paper is organized as follows. In Section 2, we will first give a brief overview of related work on ontology mining and ERP data analysis. In Section 3, we describe our generic framework for ontology mining which includes a sophisticated combination of (hierarchical) clustering, classification and association rule mining. Specifically, we apply our framework to mine an ERP domain ontology. In Section 4, we present results for ERP ontology mining based on data collected from three EEG experiments on visual word comprehension. The input to our ontology mining framework are extracted from spatiotemporal ERP data using temporal Principal Components Analysis (PCA). The mined ERP ontology is conceptually transparent for domain experts, and in particular, the mined rules (axioms) correspond closely with the labeling classification rules that were independently defined by domain experts. In Section 5, we discuss more robust data preprocessing, ontology mining, and future data integration directions for the NEMO project. Finally, in Section 6, we draw some general conclusions about our contribution regarding state-of-the-art techniques for ERP ontology mining, representation, and integration.

2. RELATED WORK

ERP data consist of time series, representing temporal fluctuations in the EEG that are time-locked to events of interest (e.g., word or picture stimuli). In dense-array EEG and ERP research, these time series are measured across multiple locations on the scalp surface. A variety of tools are available for ERP preprocessing and pattern analysis. For example, Net Station [4] is a suite of tools, which includes data cleaning, statistical extraction and visualization techniques. EEGLAB [2] is a Matlab toolbox that provides advanced statistical methods for EEG/MEG and ERP processing, including independent component analysis (ICA) and joint time-frequency analysis (TFA). APECS is a Matlab toolbox that contains tools for data cleaning (ICA and related techniques) and evaluation of data decomposition results [17]. The Dien PCA Toolbox [1] includes Principal Component Analysis (PCA) tools that are optimized for ERP data decomposition.

Ontology mining is a process for learning an ontology, including classes, class taxonomy, properties and axioms. In the existing work, researchers mainly focus on mining the ontologies from text documents (e.g., web content) [25] or other web data (web usage, web structure and web user profiles) [23]. In [28], clustering is used to discover the concepts in the ontology. Association rule mining has been adopted to discover the relationships between different concepts [26]. The NetAffx Gene ontology mining tool [11] is an interactive platform for visualizing and analyzing microarray data.

In this paper, we propose a generic framework for developing and mining domain ontologies, with specific application to the development of a first-generation ERP ontology. The target data type consists of spatiotemporal data (ERPs), and summary statistics (e.g., the "latent" or principal components that emerge from statistical analysis of ERP data). In addition to identifying classes, a hierarchy of classes and part-of relations of classes, our approach includes classification methods for mining properties and axioms (rules). This is also an important extension from our previous work [29], which focuses only on ERP pattern mining. In this paper, we first use the previous ERP pattern mining results (data from Experiment 1-2) to develop ERP classes. Furthermore, we adopt hierarchical clustering methods to generate class taxonomies and association rules to discover the property relations respectively from a new dataset (Experiment 3).

3. FRAMEWORK

Based on existing ontology mining approaches and our previous work for mining ERP patterns [29], we summarize and propose the following four general procedures for mining the concepts and their relationships in domain ontologies:

1. *Classes* $\Leftarrow$ *Clustering-based Classification*: If there exists n clusters in existing domain dataset D based on some clustering algorithm, we first define n candidate classes for the domain and assign them arbitrary names, such as "C1," "C2" for each class. After the data instances of each cluster are labeled by the assigned class name, each class (i.e., each cluster) will be formally defined by classification rules. The arbitrary class names may be updated to more meaningful ones by domain experts based on their understanding of classification rules.

2. *Class Taxonomy* $\Leftarrow$ *Hierarchical Clustering*: More granular classes and their taxonomy (hierarchy) will be determined by a hierarchial clustering algorithm. Again, we can assign arbitrary names for each new class for the clustering-based classification process. Domain experts may give more meaningful names for new classes based on classification rules.

3. *Properties* $\Leftarrow$ *Classification*: The classification process for defining rules for classes will also be used to determine candidate properties between different classes or between classes and data types.

4. *Axioms* $\Leftarrow$ *Association Mining and Classification*: The association rules between different properties will be used for defining the axioms (rules) between properties. And the classification rules will also be defined as the axioms (rules) among different classes and properties. The interaction of classification and association rule mining will be used for rule optimization (the detail will be discussed in Section 4.5.2).

All of the above four procedures and their interactions are shown in Figure 2 and the outputs (i.e., classes, class hierarchy, properties, axioms) are put together into a domain ontology. It is a semi-automatic framework because we need "expert labeling" to give meaningful names for classes. The input data are put into some semi-structured formats, such as the spreadsheet, after data preprocessing. Otherwise, some statistical or text processing step needs to be done as a part of data preprocessing.

To further explain why our ontology mining framework based on the four general procedures makes sense, we first suppose there exists a domain ontology (i.e., semantics of data) for a set of data instances in some specific domain (e.g., ERP). Our goal is to find what classes, properties

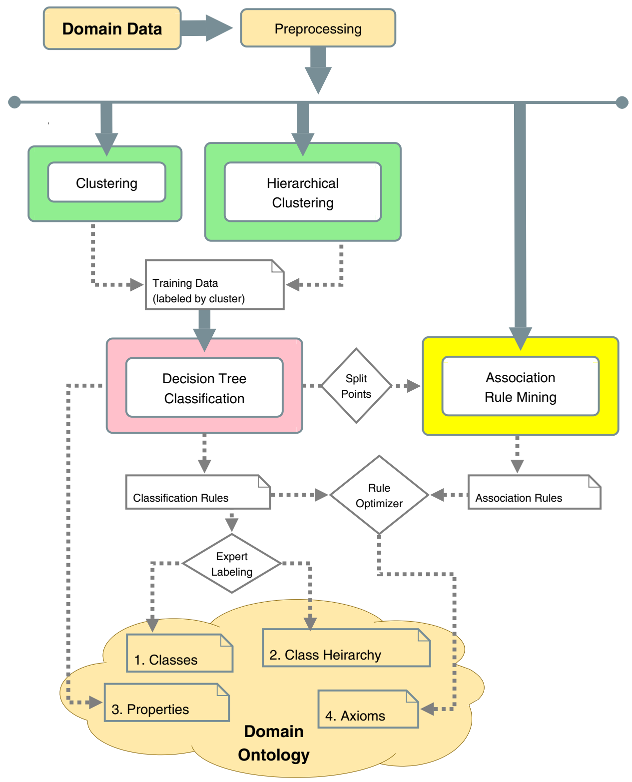

Figure 2: A semi-automatic framework for mining domain ontologies.

and axioms can be mined to compose that domain ontology. From a machine learning point of view, the domain ontology is the target function to be learned and it includes several components such as classes, class hierarchy, properties and axioms. A reasonable assumption is that the data instances which belong to the same class must be similar by sharing some properties, the data instances which belong to different classes must be dissimilar. Therefore, determining what and how many classes should be included in an ontology is typically a clustering problem. It is a natural extension that finding the hierarchy of classes (clusters) is a hierarchial clustering problem. On the other hand, what properties and values the data instances in the same class should share is a typical classification problem. The selection of attributes for classification (e.g., information gain selection) can be used for property selection in ontology mining. The classification rules can also be treated as the relationships (axioms) of properties and classes. The association rules between different properties can be treated as relationships (axioms) of two or multiple properties themselves, which will be a good complementary for the ontology.

In summary, our generic framework includes data preprocessing, clustering, hierarchical clustering, clustering-based classification and association mining. The data input into four procedures are in some semi-structured format (e.g., spreadsheet) after data preprocessing. The outputs (i.e., classes, class hierarchy, properties, axioms) will be used to compose a domain ontology. We will elaborate the details of our framework with the description of experiments of mining ERP ontologies in the following Section 4.

4. EXPERIMENTS ON ERP DATA

4.1 Data preprocessing

In this paper, we analyzed data collected in three studies of neural activity during visual word comprehension (Experiment 1 - 3). Data were acquired using a 128-channel EEG sensor net [3]. Sampling rate was 250hz. The EEG were segmented into 1,500ms epochs, beginning 500ms before stimulus onset (total number of samples = 375).

The data from Experiment 1 and Experiment 2 comprise 89 subjects and 6 experimental conditions (number of observations = 534). A description of the experiment paradigm, behavioral measures, scalp ERPs, and cortical (source) waveforms can be found in [18]. For cross-validation of our pattern classification and labeling procedures, subjects were randomly assigned to one of two groups, resulting in 20-24 subjects per subgroup. Subgroups were matched in proportion of males to females and in mean age and handedness. Data from a new experiment (Experiment 3) dataset consist of 36 subjects and 4 experiment conditions (number of observations = 144).

4.1.1 Temporal PCA decomposition

ERP data represent a mixture of "signal" (functional brain patterns) and "noise" (extracerebral artifacts and brain activity that is not related to the events of interest). Data decomposition methods can help separate signal from noise and disentangle overlapping patterns. A variety of statistical decomposition methods have been applied to ERP data in the past few decades, such as Independence Component Analysis (ICA), wavelets and Principal Component Analysis (PCA). In this paper, Principal Component Analysis [13] is used to decompose the ERP data. PCA belongs to a family of dimension reduction procedures. It projects the data into a new space of lower dimension.

In the present study, we used temporal PCA, as implemented in the Dien PCA Toolbox [1]. The dataset used as input to the PCA is organized with the variables corresponding to time points. The number of variables is equal to the number of samples (number = 375). The waveforms vary across subjects, channels (number = 128) and experimental conditions. PCA extracts as many factors as there are variables. After rotation of extracted factors, a small subset of the factors are retained for further analysis. In this experiment, we retained the first 15 PCA factors, accounting for most of the variance (> 75%). The remaining factors are assumed to contain "noise." This assumption is verified by visual inspection of the time course and topographic projection of each factor.

4.1.2 Summary metrics extraction

For each PCA factor, we extracted summary metrics representing spatial, temporal and functional dimensions of the ERP patterns of interest. After preprocessing, the Experiment 1 and Experiment 2 datasets consist of vectors containing 25 spatial, temporal and functional attributes derived from the automated measure generation. Thus, the data represent the individual PCA factors of each subject and condition as points in a 25 dimensional attribute space. For the Experiment 3 dataset, we increase the number of attributes to 31 by adding more intensity attributes, such as "Pseudo-Known" (Difference in mean intensity over region of interest at time of peak latency (Nonwords-Words)).

Attribute	Description
IN-min	min amplitude
IN-max	max amplitude
IN-mean	mean amplitude for a specified channel set
ROI	region of interest
SP-cor	cross-correlation between Factor(FA) topography and topography of target pattern
SP-max	channel with max weighting for factor FA
SP-max (ROI)	channel grouping(ROI) to which the max channel belongs
SP-min	channel with min weighting for factor FA
SP-min(ROI)	channel grouping(ROI) to which the min channel belongs
TI-max	max latency(time of max amplitude)
EVENT	event type (stimon, respon, EKG-R, etc.)
STIM	stimulus
MOD	modality of stimulus

Table 1: Intensity, spatial, temporal and functional metrics

Table 1 lists some common attributes that are used for all the datasets. The datasets were put into spreadsheets and each column corresponds to an attribute. Domain experts labeled 3 kinds of pattern factors for Experiment 1 group 1 data, 4 for Experiment 1 group 2 data and 8 for Experiment 3 data. For example, four spatiotemporal patterns relating to visual object processing are: the P100 (an occipital positivity, peaking at 100ms), N100 (an occipital negativity, peaking at 180ms), N2 (a left temporal pattern, peaking at 250ms), and P300 (a parietal positivity from 300 to 700ms).

4.2 Mining ERP Classes with Clustering

Traditionally, ERP patterns are identified through visual inspection of grand-averaged ERP data. However, the precise definition of a target pattern, its operationalization, and measurement across individual subjects, can vary considerably across research groups. In our framework, we use clustering to automatically separate ERP patterns, as they are distributed across "latent" (PCA) factors. The factors extracted through PCA are weighted across individual subjects and experiment conditions. Summary metrics extracted from each observation (subject and condition) are then input into clustering tool. Observations that belong to the same pattern are expected to map to the same cluster using this method. The larger aim is to develop an automatic pattern classification method, which can support robust ERP pattern definitions.

4.2.1 *Expectation-Maximization clustering*

The Expectation-Maximization (EM) algorithm [12] is often used to approximate distributions using mixture models. It is an iterative procedure that circles around the expectation and maximization steps (i.e., E-step and M-step). EM clustering can assign each object to a cluster according to a weight representing the probability of membership. The goal is to "maximize" the likelihood of the distributions given the data. We also tried other classical clustering algorithms, such as K-Means and K-Medoids. It seems EM works better than others, especially in the scenario that the number of clusters (e.g., ERP patterns) is indefinite.

Cluster/Pattern	0	1	2
P100	0	0	**99**
N100	46	47	0
lateN1/N2	47	**235**	0

Table 2: EM Clustering Results for Experiment 1 group 1 Pattern Factors

Cluster/Pattern	0	1	2	3
P100	0	**76**	0	2
N100	**117**	1	0	54
lateN1/N2	13	14	0	**104**
P300	0	61	**110**	42

Table 3: EM Clustering Results for Experiment 1 group 2 Pattern Factors

In the E-step for clustering, the algorithm calculates the posterior probability that a data instance (e.g., a data tuple with 25 attributes in our ERP experiment) belongs to a cluster. In the M-step, EM algorithm searches for optimal parameters that maximize the sum of log-likelihood probabilities. EM algorithm automatically selects the number of clusters by maximizing the logarithm of the likelihood of future data. The detailed implementation of EM clustering can be found at [21]. And we use EM clustering algorithm in WEKA [9] in the experiments.

4.2.2 *Clustering results*

For each of the experimental datasets, we applied EM clustering to the summary metrics described previously in Table 1. In the current study, the data in each cluster were compared with the human labeling result (which are generated with the rules defined by domain experts) to determine the distribution of the pre-defined ERP patterns amongst the clusters. The number of clusters was set equal to the number of patterns that were identified by domain experts. Observations were then assigned to clusters using this semi-automatic approach. Table 2, 3 and 4 show the clustering results for Experiment 1 group 1, Experiment 1 group 2 and Experiment 3 data. The resulting assignment of observations to clusters corresponded closely with the pattern labeling results based on expert judgments. Compared with our generic framework shown in Figure 2, domain experts actually did "expert labeling" for all data instances before the clustering step. However, we did not input the

C/P	0	1	2	3	4	5	6	7
P100	0	1	0	**109**	0	0	0	0
N100	0	0	20	0	8	**85**	2	0
N300	0	0	34	0	14	1	5	0
lateN1/N2	0	0	0	0	49	4	**79**	0
P1r	0	0	**76**	0	16	0	9	0
MFN	0	25	0	0	0	0	0	**40**
N400	0	9	0	0	0	0	0	7
P300	**108**	5	0	0	0	0	0	2

Table 4: EM Clustering Results for Experiment 3 Pattern Factors. "C/P" means "Cluster/Patten."

labels into the EM clustering. Instead, we only use them to compare with clustering result and replace arbitrary cluster (class) names by corresponding patter names. In more general cases, we believe that "expert labeling" can only happen with the help of discovered classification rules. The data instances with labels do not always exist before the clustering step. On the other hand, there was not a strict one-to-one mapping between clusters and labeled patterns. Rather, the results showed some pattern "splitting," where observations belonging to a target pattern were assigned to more than one cluster. The proper diagnosis and interpretation of such results will require careful system evaluation to determine the source of this "misallocation of variance."

Based on the clustering result, we can generate the following OWL classes:

```
<rdf:RDF
 xmlns:time="http://www.isi.edu/~pan/damltime/time.owl#"
 xmlns:owl="http://www.w3.org/2002/07/owl#">

<owl:Class rdf:ID="ERPPattern">
 <rdfs:subClassOf
        rdf:resource="time#TemporalEntity"/>
</owl:Class>

<owl:Class rdf:ID="P100">
 <rdfs:subClassOf
        rdf:resource="#ERPPattern"/>
</owl:Class>

<owl:Class rdf:ID="N100">
 <rdfs:subClassOf
        rdf:resource="#ERPPattern"/>
</owl:Class>

<owl:Class rdf:ID="LateN1/N2">
 <rdfs:subClassOf
        rdf:resource="ERPPattern"/>
</owl:Class>

<owl:Class rdf:ID="P300">
 <rdfs:subClassOf
        rdf:resource="ERPPattern"/>
</owl:Class>
 ...
```

4.3 Mining ERP Class Taxonomy with Hierarchical Clustering

The Experiment 1 and Experiment 2 only include 3 or 4 patterns (classes) and it is hard to show how to mine another important component, class taxonomy, for an ERP ontology. To show that, we analyzed the Experiment 3 data. According to prior ERP research, we would expect to find about 8 patterns between 100 to 700ms after presentation of a visual word stimulus. These patterns include the P100, N100, lateN1/N2, N3, MFN, N400, P1r, and P300. We expect the hierarchical clustering to discover these patterns automatically, and also to generate a taxonomy of these patterns.

4.3.1 *Methodology*

We have applied EM clustering in a hierarchical way to discover the class taxonomy, using both divisive and agglomerative strategies. In the divisive approach, we first put all the data from 8 ERP patterns (classes) into one cluster. Our goals is to sub-divide this cluster into 2 clusters by setting up the number of clusters. Then we repeatedly sub-divide each cluster until the majority of data instances from each pattern forms a cluster.

In each step, the data instances of a particular pattern (class) labeled by domain experts may go to different clusters. We always keep the majority of data instances for each pattern in one cluster but take out those in other clusters. In agglomerative approach, we first put the data instances into 8 clusters to reflect expert hypotheses regarding the number of distinct patterns. Then we try to merge them into 7 clusters by setting up the number of clusters. Again, we only keep the majority data instances for each pattern. We continue to merge them into fewer clusters until all data instances can be put into one cluster, if possible.

4.3.2 *Clustering results*

For the eight patterns our neuroscientists want to discover from the data, both divisive and agglomerative clustering approaches result in the same hierarchy shown in Figure 3.

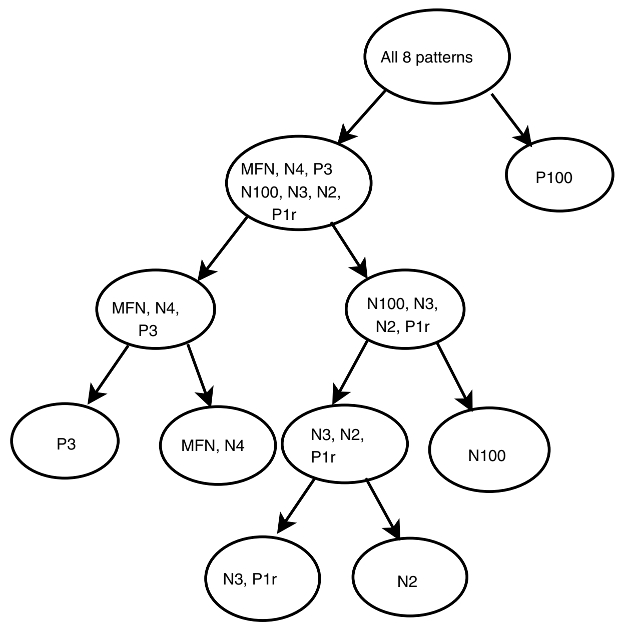

Figure 3: The hierarchy graph of 8 ERP patterns (classes), where "P3" means P300, "N2" means lateN1/N2, "N3" means N300, "N4" means N400.

The hierarchy in Figure 3 shows that the MFN and N4 patterns belong to the same cluster (class). Likewise, the N3 and P1r patterns belong to the same cluster. These results suggest one of two possibilities. First, it is possible that patterns previously assigned distinct labels in the ERP literature reflect one and the same underlying process. Second, it is possible that these patterns are in fact distinct, but our analyses failed to separate them. In this second case, it will be important to refine human labeling steps to capture fine-grained distinctions between spatiotemporal patterns. Finally, it will be critical to include data collected across a range of experiment paradigms, to provide a broader range of functional data that can be used for pattern analysis.

The discovered hierarchy (class taxonomy) can be represented in OWL and added into the ERP ontology like:

```
<owl:Class rdf:ID="MFN/N4/P3/N100/N3/N2/P1r">
 <rdfs:subClassOf
        rdf:resource="#ERPPattern"/>
</owl:Class>
```

```
<owl:Class rdf:ID="MFN/N4/P3">
 <rdfs:subClassOf
        rdf:resource="MFN/N4/P3/N100/N3/N2/P1r"/>
</owl:Class>

<owl:Class rdf:ID="MFN/N4">
 <rdfs:subClassOf
        rdf:resource="MFN/N4/P3"/>
</owl:Class>

<owl:Class rdf:ID="P300">
 <rdfs:subClassOf
        rdf:resource="MFN/N4/P3"/>
</owl:Class>
  ...
```

4.4 Mining Properties and Axioms (Rules) with Clustering-based Classification

The EM clustering process can partition pattern factors into several clusters (i.e., OWL classes) in a hierarchical way, such that each cluster is mainly comprised of one or several categories of pattern factors. Our next goal is to discover the axioms (rules), which specify the properties and their relationships with defined classes.

4.4.1 Methodology

After EM clustering and hierarchical clustering, we use C4.5 classification algorithm [27] to build a decision tree to classify factors in each cluster. C4.5 is a standard decision learning algorithm which works well for continuous values. Some ERP attributes have continuous values. On the other hand, the classification rules derived from the decision tree are meaningful to human experts. The discovered rules will be used for defining the axioms in the ERP ontology which specify the properties and their relationships with defined classes.

Considering the number of clusters needs to be referred to the labeling efforts of domain experts, the current process is semi-automatic for mining ERP ontologies. Once the data mining process becomes more robust, we will not need the data labeling effort from domain experts. But we will need domain experts to give meaningful names for classes based on the classification rules.

We use J48 in WEKA, which is an implementation of the C4.5 algorithm, to classify the data. The input of the decision tree classifier is the pattern factor metrics vector and their arbitrary cluster names are used as classification labels. We built decision trees based on the clusters discovered from Experiment 1 and Experiment 2 data, and also the cluster hierarchy from Experiment 3 data. We use information gain [21] as an attribute selection measure in building the decision tree. It always chooses the attribute that is most capable of differentiating different classes of data at each level of the tree. Those selected attributes will be considered as properties in ERP ontology.

4.4.2 Classification results

Figure 4 shows the decision tree learner trained on Experiment 1 group 1 data. It achieves a precision of 97.44% on the training data.

From Figure 4, we can see that although 25 attributes are input to the learning process, only 6 of them are used in the final decision tree classifier. For instance, Table 5 is a

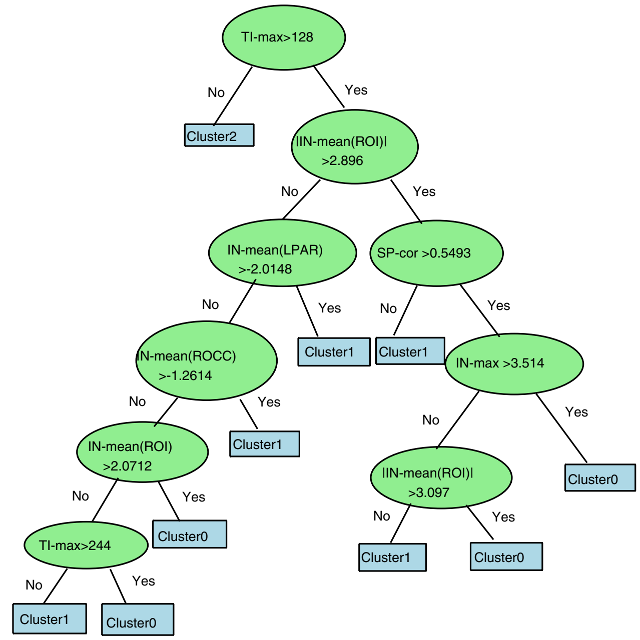

Figure 4: Decision tree classifier.

Attribute	Average-Merit	Average Ranking
TI-max	0.836	1
IN-mean (ROI)	0.238	2.2
IN-mean (ROCC)	0.224	3.3
SP-cor	0.215	3.6
...	...	...

Table 5: Information gain of summary metrics

table of information gain of attributes. In the rules provided by domain experts, only TI-max and IN-mean (ROI) are used. However, information gains of the attributes show that IN-mean (ROCC) is also important in the classification of patterns.

Therefore, we consider the top attributes in Table 1 as the candidate properties. The information gain selection help us (including domain experts) to determine which properties should be in the ERP ontology. For example, in OWL, a subset of properties can be represented as

```
<owl:DatatypeProperty rdf:ID="IN-mean(ROI)_maxvalue">
  <rdfs:domain rdf:resource="#ERPFactor"/>
  <rdfs:range rdf:resource="xsd#float" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="IN-mean(ROI)_minvalue">
  <rdfs:domain rdf:resource="#ERPFactor"/>
  <rdfs:range rdf:resource="xsd#float" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="TI-max_maxvalue">
  <rdfs:domain rdf:resource="#ERPFactor"/>
  <rdfs:range rdf:resource="xsd#integer" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="TI-max_minvalue">
  <rdfs:domain rdf:resource="#ERPFactor"/>
  <rdfs:range rdf:resource="xsd#integer" />
</owl:DatatypeProperty>
```

Expert-defined rule	Decision tree rule
$\forall n,\ FAn = N100\ \ if$ $150 < TI-max <= 220$ $\wedge IN-mean(ROI) < -0.4$ $\wedge EVENT\ =\ stimon$ $\wedge MODALITY\ =\ visual$	$\forall n,\ FAn \in cluster0\ \ if$ $TI-max > 128$ $\wedge \lvert IN-mean(ROI)\rvert > 2.896$ $\wedge SP-cor > 0.549$ $\wedge IN-max > 3.514$
$\forall n,\ FAn = lateN1/N2\ \ if$ $220 < TI-max <= 300$ $\wedge IN-mean(ROI) < -0.4$ $\wedge EVENT\ =\ stimon$ $\wedge MODALITY\ =\ visual$	$\forall n,\ FAn \in cluster1\ \ if$ $TI-max > 128$ $\wedge \lvert IN-mean(ROI)\rvert > 2.896$ $\wedge SP-cor <= 0.549$

Table 6: Expert-defined rules vs. Decision tree generated rules from Experiment 1 group 1 dataset

4.4.3 Rule comparison with domain experts

One advantage of using decision tree is that we can generate rules from decision tree and compare them with the ones that are defined by domain experts. It can help domain experts to determine (i.e.,"expert labeling") the names of clusters if their names were arbitrarily assigned. Table 6 compares the rules for N100 and lateN1/N2 patterns, where cluster0 and cluster1 correspond to N100 and lateN1/N2 respectively. From it, we can see that decision tree uses more attributes. The values for EVENT and MODALITY are the same for all the data in Experiment 1 group 1. Therefore EVENT and MODALITY are not selected by decision tree classifier. On the other hand, the attribute values of decision tree rules can not be exactly the same as domain expert rules, although they are basically consistent. We believe the attributes that are used in the decision tree and their values can be a good reference for domain experts to refine their rules.

4.4.4 Rule representation in ERP ontologies

The classification rules derived from the decision tree will be used to determine what axioms in the ontology can describe the relationships between properties and classes. However, what will be a standard logic language for rules is still an open question in the Semantic Web research. SWRL [7] can be a choice and it is a subset of first order logic. For example, the decision tree rule related to cluster1 (corresponding to lateN1/N2) in Table 6 can be represented in SWRL like:

```
<ruleml:Imp>
  <ruleml:body rdf:parseType="Collection">
   <swrlx:classAtom>
     <owlx:Class owlx:name="&erp;Factor" />
     <ruleml:var>f</ruleml:var>
   </swrlx:classAtom>
   <swrlx:datavaluedPropertyAtom
           swrlx:property="TI-max_minvalue">
     <swrl:argument1 rdf:resource="#f" />
     <owlx:DataValue owlx:datatype="&xsd;int">
                         128</owlx:DataValue>
   </swrlx:datavaluedPropertyAtom>
   <swrlx:datavaluedPropertyAtom
           swrlx:property="IN-mean(ROI)_minvalue">
     <swrl:argument1 rdf:resource="#f" />
     <owlx:DataValue owlx:datatype="&xsd;int">
                            2.896</owlx:DataValue>
   </swrlx:datavaluedPropertyAtom>
   ...
  </ruleml:body>
  <ruleml:head rdf:parseType="Collection">
   <swrl:IndividualPropertyAtom>
     <swrl:propertyPredicate rdf:resource="#labeled_as"/>
     <swrl:argument1 rdf:resource="#f" />
     <swrl:argument2 rdf:resource="#LateN1/N2" />
   </swrl:IndividualPropertyAtom>
  </ruleml:head>
</ruleml:Imp>
```

In this paper, to save the space, we may just use general first order logic axioms to represent the above SWRL rule. It looks like:

$$\begin{aligned}&\forall f\ Factor(f)\ \wedge\ TI-max\_minvalue(f,128)\\&\quad \wedge\ IN-mean(ROI)\_minvalue(f,2.896)\\&\quad \wedge\ SP-cor\_maxvalue(f,0.549)\\&\quad\quad \rightarrow labeled\_as(f,LateN1/N2)\end{aligned}$$

4.5 Discovering Axioms among Properties with Association Rules Mining

Clustering and classification methods induce the axioms to distinguish different patterns (classes) using properties. Relationships between properties themselves is also of interests of domain experts and can be put as axioms into the ERP ontology. In this paper, we use association rule mining to discover the relationships between different properties.

Association rule
(1) $IN-mean(ROI) < 0.9243 \Rightarrow IN-LATEM > -2.0743$
(2) $IN-RORB <= 0.70 \Rightarrow IN-LFRON <= 2.049$
(3) $SP-cor > 0.2295 \Rightarrow IN-LATEM > -2.0743$
(4) $SP-cor > 0.2295 \wedge RareMisses-RareHits < 0.5154$ $\Rightarrow IN-LATEM > -2.0743$

Table 7: Association rules from Experiment 3 dataset

4.5.1 Methodology

Association rule mining aims at finding frequent patterns in certain data sets. In our case, association rule mining is used to seek the properties that frequently co-occur for the specific ERP pattern factors. After decision tree classification, we quantify the values of each attribute by using their splitting point value in the tree. This converts the numeric values of each attribute to categorical values. Then, we applied the well-used Apriori algorithm [10] in Weka [9] to find association rules of these attributes. Table 7 lists a subset of the association rules we generated. We only selected those association rules with high confidence (i.e., >90%) and put them into the ERP ontology.

Association rules should also be represented as logic (e.g., SWRL) axioms in the ERP ontology. For example, the first and fourth association rules in Table 7 can be represented in general first order axioms:

$$\begin{aligned}&(1)\ \forall f\ Factor(f)\ \wedge\ IN-mean(ROI)\_maxvalue(f,0.9243)\\&\quad\quad \rightarrow IN-LATEM\_minvalue(f,-2.0743)\end{aligned}$$

$$\begin{aligned}&(4)\ \forall f\ Factor(f)\ \wedge\ SP-cor\_minvalue(f,0.2295)\\&\quad \wedge\ RareMisses-RareHits\_maxvalue(f,0.5154)\\&\quad\quad \rightarrow IN-LATEM\_minvalue(f,-2.0743)\end{aligned}$$

4.5.2 Rule optimization

As the number of ERP patterns and attributes increases, the decision tree generated by classification step expands. The path from root node to the leaf node in a tree, which corresponds to one rule for a cluster, becomes longer. However, we can use association rules to trim the classification rules. An important inference rule [30] in logic is:

$$(A \to B) \wedge (B \wedge A \ \to \ C) \Rightarrow (A \to \ C)$$

It can be applied when the parameters (e.g., A and B) in the condition of a classification rule (e.g., $B \wedge A \ \to \ C$) are closely related to an association rule (e.g., $A \to B$ or $B \to A$). For instance, there is one rule for cluster4, which corresponds to N3 pattern, from the Experiment 3 dataset:

$$\begin{aligned}
&\forall f \ Factor(f) \ \wedge \ TI - max_minvalue(f, 102) \\
&\quad \wedge \ TI - max_maxvalue(f, 204) \\
&\quad \wedge \ SP - cor_minvalue(f, 0.2295) \\
&\quad \wedge \ IN - min_to_Baseline_minvalue(f, -4.2760) \\
&\quad \wedge \ SP - max_maxvalue(f, 46) \\
&\quad \wedge \ RareMisses - RareHits_maxvalue(f, 0.5154) \\
&\qquad \to labeled_as(f, N3)
\end{aligned}$$

We can use the fourth association rule in Table 7 to optimize the above classification rule to:

$$\begin{aligned}
&\forall f \ Factor(f) \ \wedge \ TI - max_minvalue(f, 102) \\
&\quad \wedge \ TI - max_maxvalue(f, 204) \\
&\quad \wedge \ SP - cor_minvalue(f, 0.2295) \\
&\quad \wedge \ IN - min_to_Baseline_minvalue(f, -4.2760) \\
&\quad \wedge \ SP - max_maxvalue(f, 46) \\
&\qquad \to labeled_as(f, N3)
\end{aligned}$$

We have built an ontology inference engine, OntoEngine, which can be extended to implement this kind of optimization (transformation) for axioms [16]. The inference process will continue until no rule can be trimmed anymore.

5. DISCUSSION AND FUTURE WORK

In this paper, we have outlined a new framework for mining ERP ontologies based on clustering, classification and association rule mining. Our first-generation ERP ontology consists of 16 classes, 57 properties and 23 axioms. We show a partial view of this preliminary ERP ontology in Figure 5.

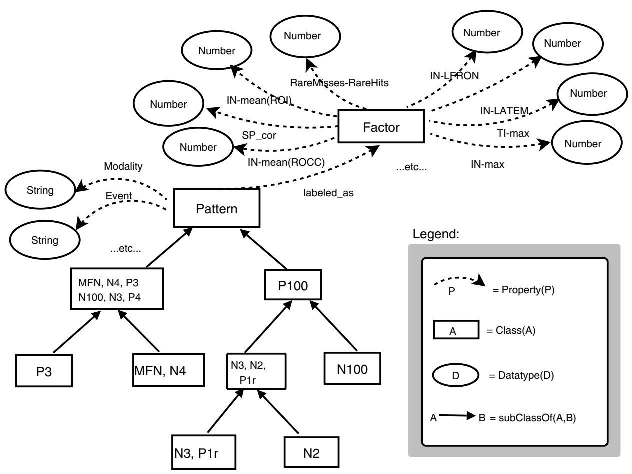

Figure 5: A partial view of a mined ERP ontology.

Figure 5 shows 10 classes, i.e., factor and pattern taxonomy. Patterns have temporal, spatial and functional attributes (some of which are listed in the graph, such as Event, Modality etc.) which are represented as properties of the "Pattern" class in the ERP ontology. TI-max, IN-mean(ROI) etc are properties of "Factor" which have number values. "Factor" relates to "Pattern" by a "labeled_as" property.

As described here, this approach can be highly informative when applied to PCA-based metrics generated from high-density ERP data. Part of the ongoing work is focused on further refinements to our clustering process. For example, in the present set of experiments, some patterns "split" across (were assigned to) more than one cluster. Inspection of temporal PCA results suggested that refinements to the data decomposition process, as well as additional metrics that capture temporal and spatial attributes more accurately, may reduce this "misallocation" of pattern variance. To achieve accuracy in system evaluation, we will compare system results with a "gold standard," which will be established by expert labeling of early visual-evoked ERP patterns (e.g., P100v, N100v, and N2v).

To show our ontology mining framework is generic and robust, we will apply a variety of data preprocessing techniques besides PCA decomposition. It will also be interesting to try our ontology mining framework in a range of experimental paradigms, including auditory as well as visual stimulus processing, and nonlinguistic as well as language-related paradigms.

Another important aim of the NEMO project is to store high-level pattern descriptions in an ERP *ontology database*, which is automatically modeled based on the semantics of an ERP ontology. The next phase of the NEMO project will be focused on development of an ontology-based integration system which will facilitate the representation and dissemination of ERP data across studies and labs. Different labs will create their own ERP ontologies and ontology databases. Ontology-based integration in NEMO will study semantic mapping rules between different ERP ontologies. Given the mapping rules, once the user query comes in, various ERP ontology databases can be searched for answers to the query. We reported an efficient ontology-based data integration system (OntoGrate) in [15], which will be extended to support NEMO. We will implement the data exchange and query answering components through the inference engine by reasoning with ERP ontologies and mapping rules.

In general, we expect that this ontology-based methodology can be extended for integrating other types of neuroscience data (e.g., fMRI data) and support other biomedical data sharing efforts (e.g., the Gene Ontology).

6. CONCLUSION

In this paper, we introduce a generic framework for mining domain ontologies and present some results of our work on development of a first-generation ERP ontology. This work aims at exploring methods for differentiating different ERP pattern factors and selecting important concepts and rules for ontology definition.

The framework works well for ERP ontology definition. The raw ERP data can be preprocessed to ERP factors by temporal Principal Component Analysis. EM-based clustering and hierarchical clustering can cluster ERP factors to different groups (i.e., classes and class taxonomy in the ontology). Then we use classification method (C4.5 decision tree learning) to get ERP pattern rules automatically after labeling the factors with the result from clustering. The classification rules are consistent with domain experts' rules and can be used as references for further human refinement. The classification rules are used to define the axioms between properties and classes in the ERP ontology. Association rule mining are used to define axioms among properties and to help rule optimization.

The future work of the NEMO project is to make our system more robust in different experimental paradigms. The design of ontology databases and the discovery of their mappings will support data integration across different ERP laboratories. We expect that our NEMO framework will be extended to other types of neuroscience data and to support other biomedical ontology-based data sharing efforts.

Acknowledgements

We thank Paea LePendu for his input to the final version of this paper. We also thank the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] Dien, J. PCA Toolbox (version 1.7). `http://people.ku.edu/~jdien/`.

[2] EEG lab. `http://www.sccn.ucsd.edu/eeglab/`.

[3] Electrical Geodesics, Inc. `http://www.egi.com`.

[4] NetStation Technical Manual. `http://www.egi.com`.

[5] Open Knowledge Base Connectivity (OKBC). `http://www.ai.sri.com/~okbc/`.

[6] OWL Web Ontology Language. `http://www.w3.org/TR/owl-ref/`.

[7] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. `http://www.w3.org/Submission/SWRL/`.

[8] The National Center for Biomedical Ontology. `http://www.bioontology.org/`.

[9] Weka 3: Data Mining Software in Java. `http://www.cs.waikato.ac.nz/ml/weka/`.

[10] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Very Large Data Bases (VLDB) Conference*, pages 487–499. Morgan Kaufmann, 1994.

[11] J. Cheng, S. Sun, A. Tracy, E. Hubbell, J. Morris, V. Valmeekam, A. Kimbrough, M. S. Cline, G. Liu, R. Shigeta, D. Kulp, and M. A. Siani-Rose. Netaffx gene ontology mining tool: a visual approach for microarray data analysis. *Bioinformatics*, 20(9):1462–1463, 2004.

[12] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society*, 39:1–38, 1977.

[13] J. Dien. Addressing misallocation of variance in principal components analysis of event-related potentials. *Brain Topography*, 11(1):43–55, January 1998.

[14] J. Dien and G. Frishkoff. Introduction to principal components analysis of event-related potentials. pages 189–208. MIT Press, 2005.

[15] D. Dou, P. LePendu, S. Kim, and P. Qi. Integrating Databases into the Semantic Web through an Ontology-based Framework. In *Proceedings of the third International Workshop on Semantic Web and Databases (SWDB'06)*, page 54, 2006. co-located with ICDE 2006.

[16] D. Dou and D. McDermott. Deriving Axioms Across Ontologies. In *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06) (short paper)*, pages 952–954.

[17] R. M. Frank and G. A. Frishkoff. Automated protocol for evaluation of electromagnetic component separation (apecs): Application of a framework for evaluating statistical methods of blink extraction from multichannel eeg. *Clin Neurophysiol*, 118(1):80–97, 2007.

[18] G. A. Frishkoff. Hemispheric differences in strong versus weak semantic priming: Evidence from event-related brain potentials. *Brain & Language*, 100(1):23–43, 2007.

[19] T. Gene Ontology Consortium. Creating the Gene Ontology Resource: Design and Implementation. *Genome Research*, 11(8):1425–1433, 2001.

[20] M. R. Genesereth. Knowledge Interchange Format (KIF). `http://logic.stanford.edu/kif/kif.html`.

[21] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[22] D. B. Keator, S. Gadde, J. S. Grethe, D. V. Taylor, and S. G. Potkin. A general xml schema and spm toolbox for storage of neuro-imaging results and anatomical labels. *Neuroinformatics*, 4(2):199–212, 2006.

[23] Y. Li and N. Zhong. Mining ontology for automatically acquiring web user information needs. *IEEE Trans. Knowl. Data Eng.*, 18(4):554–568, 2006.

[24] D. Lindberg, B. Humphries, and A. McCray. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291, 1993.

[25] A. Maedche and S. Staab. Mining ontologies from text. In *Proceedings of International Conference on Knowledge Engineering and Knowledg Management (EKAW)*, pages 189–202, 2000.

[26] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

[27] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[28] M.-L. Reinberger, P. Spyns, W. Daelemans, and R. Meersman. Mining for lexons: Applying unsupervised learning methods to create ontology bases. In *Proceedings of International Conference on Ontologies, Databases and Application of Semantics (ODBASE)*, pages 803–819, 2003.

[29] J. Rong, D. Dou, G. Frishkoff, R. Frank, A. Malony, and D. Tucker. A Semi-automatic Framework for Mining ERP Patterns. In *Proceedings of The IEEE International Symposium on Data Mining and Information Retrieval (DMIR).*, volume 1, pages 329–334, 2007.

[30] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc, 1995.