A Lexical Knowledge Representation Model for Natural Language Understanding

Ping Chen, University of Houston-Downtown, USA

Wei Ding, University of Massachusetts-Boston, USA

Chengmin Ding, IBM Business Consulting, USA

ABSTRACT

Knowledge representation is essential for semantics modeling and intelligent information processing. For decades researchers have proposed many knowledge representation techniques. However, it is a daunting problem how to capture deep semantic information effectively and support the construction of a large-scale knowledge base efficiently. This paper describes a new knowledge representation model, SenseNet, which provides semantic support for commonsense reasoning and natural language processing. SenseNet is formalized with a Hidden Markov Model. An inference algorithm is proposed to simulate human-like natural language understanding procedure. A new measurement, confidence, is introduced to facilitate the natural language understanding. The authors present a detailed case study of applying SenseNet to retrieving compensation information from company proxy filings.

Keywords: *Computational Semantics, Hidden Markov Model, Information Retrieval, Knowledge Representation, Natural Language Processing*

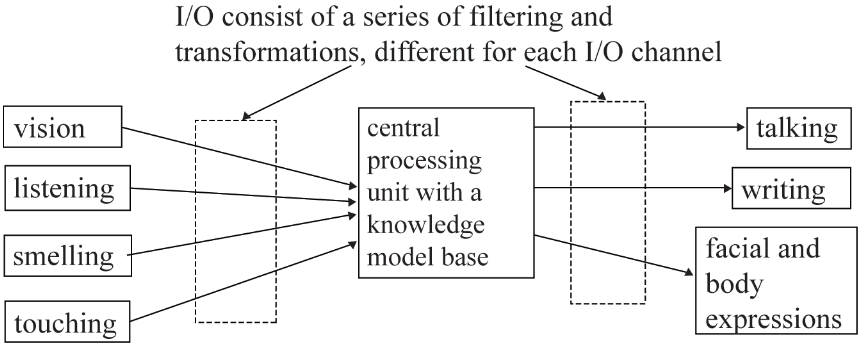
INTRODUCTION

A natural language represents and models information of real world entities and relations. There exist a large number of entities in the world, and the number of relations among entities is even higher. Entities and relations together make a highly complex multiple dimensional lattices. It is not a surprise that it usually takes a lot of training for a human being to speak, write and understand a natural language even with the fact that the computation power packed in a small human brain surpasses the most powerful supercomputer in many aspects.

Human beings receive information through vision, hearing, smelling and touching, and send information through facial and body expressions, talking and writing. Of these communication channels, reading (from human vision), hearing, talking and writing are related to natural languages. All of them are temporally one-dimensional, and only one signal is sent out or received at a certain time point, so a natural language is communicated one dimensionally.

DOI: 10.4018/jssci.2009062502

Figure 1. Communication process for a knowledge-based system


I/O consist of a series of filtering and transformations, different for each I/O channel
vision
listening
smelling
touching
central processing unit with a knowledge model base
talking
writing
facial and body expressions

With one-dimensional natural languages used by human being, in order to understand and describe a highly dimensional environment a series of filtering and transformations are necessary as illustrated in Figure 1. These transformations can be N-dimensional to N-dimensional or one-dimensional to N-dimensional in input process, and N-dimensional to one-dimensional or N-dimensional to N-dimensional in an output process. After these transformations information should be ready to be used by the central processing unit directly. Effectiveness and efficiency of these transformations are very important to knowledge representation and management.

A knowledge model describes structure and other properties of a knowledge base which is part of a central processing system. A knowledge representation model is simply a mirror of our world, since one important requirement for a model is its accuracy. In this sense there is hardly any intelligence in a knowledge model or a knowledge base. Instead it is the communication process consisting of filtering and transformations that shows more intelligent behaviors. As expressed by Robert C. Berwick, et al., in a white paper of MIT Genesis project (Berwick, et. al., 2004), "The intelligence is in the I/O". As shown in Figure 1, a knowledge model may be the easiest component to start since its input has been filtered and transformed tremendously from the original format, and is ready to be stored in the knowledge base directly. On the other hand, a knowledge representation (KR) model plays a central role to any knowledge-based systems, and it eventually decides how far such a system can go. Furthermore, knowledge and experience can make the process of filtering and transformations more efficient and effective.

A KR model captures the properties of real world entities and their relationships. Enormous amounts of intervened entities constitute a highly complex multi-dimensional structure. Thus a KR method needs powerful expressiveness to model such information.

Many cognitive models of knowledge representation have been proposed in cognitive informatics. Several cognitive models are discussed in (Wang & Wang, 2006). Object-Attribute-Relation model is proposed to represent the formal information and knowledge structures acquired and learned in the brain (Wang, 2007). This model explores several interesting physical and physiological aspects of brain learning and gives a plausible estimation of human memory capability. The cognitive foundations and processes of consciousness and attention are critical to cognitive informatics. How abstract consciousness is generated by physical and physiological organs are discussed in (Wang & Wang 2008). A nested cognitive model to explain the process of reading Chinese characters is presented in (Zheng, et. al., 2008), which indicates that there are two distinctive pathways in reading Chinese characters, and this can be employed to build reading models. Visual semantic algebra (VSA), a new form of denotational mathematics, is presented for abstract visual object and architecture manipulation (Wang, 2008). VSA can serve as a powerful man-machine interactive language for representing and manipulating visual geometrical objects in computational intelligence systems.

In Artificail Intelligence many KR techniques have been proposed since 1960's, such as semantic network, frame, scripts, logic rules etc. However, we still know little about how to capture deep semantic information effectively and support the construction of a large-scale commonsense knowledge base efficiently. Previous research focuses more on the expressiveness of KR. Recently there is an emerging interest of how to construct a large-scale knowledge base efficiently. In this paper we present a new KR model, *SenseNet*, which provides semantic support for commonsense reasoning and natural language understanding.

Our Contributions

SenseNet shares the same goal of building a large-scale commonsense knowledge base. Compared with WordNet, Cyc, and ConceptNet, our contributions are:

- We use a sense instead of a word as the building block for SenseNet, because a sense encodes semantic information more clearly.
- A relationship is defined as a probability matrix, which allows adaptive learning and leads naturally to human-like reasoning.
- Relationships among senses are formalized with a Hidden Markov Model (HMM), which gives SenseNet a solid mathematical foundation.
- A new measurement, confidence, is introduced to facilitate natural language understanding procedure.
- After the regular learning, SenseNet uses a "thinking" phase to generate new knowledge.

This paper is organized as follows. Section 2 discusses related work. We present our KR model, SenseNet, in section 3 and its inference algorithm in section 4. Section 5 shows how SenseNet can be used to model the human communication process. Section 6 describes a real world application on information extraction. Finally we conclude in section 7.

RELATED WORK

Knowledge Acquisition

A lot of research on building general-purpose or commonsense knowledge bases has recognized the importance of representing relations among words. Here we will discuss three major knowledge acquisition projects, Cyc, WordNet and ConceptNet.

WordNet is a widely used semantic resource in computational linguistics community (Fellbaum, 1998). It is a database of linked words, primarily nouns, verbs, adjectives and adverbs. These words are organized into synonym sets called synsets, and each synset represents one lexical concept. Meanings of each word are organized into "senses". Links are predefined semantic relations among words, not senses. Currently WordNet contains about 150,000 words/strings, 110,000 synsets and 200,000 word-sense pairs. Predefined relations can only satisfy some applications or domains no matter how carefully they are chosen, also lack of adaptiveness limits its learning capability.

The Cyc project emphasizes on formalization of commonsense knowledge into a logical framework (Witbrock, Baxter, & Curtis, 2003). Same as WordNet, its knowledge base is handcrafted by knowledge engineers. To use Cyc a natural language has to be transformed to a proprietary logical representation. Although a logical foundation has some nice properties, it is complex and expensive to apply Cyc to practical textual mining tasks.

ConceptNet is proposed in Open Mind Common Sense project in MIT. Comparing with WordNet and Cyc, the main advantage of ConceptNet is its unique way to acquire knowledge. Thousands of common people contribute through the Web by inputting sentences in a fill-in-the-blank fashion. Then concepts and binary-relational assertions are extracted to form ConceptNet's semantic network. At present ConceptNet contains 1.6 million edges connecting more than 300,000 nodes (Liu & Singh, 2004). Nodes are semi-structured English fragments, interrelated by an ontology of twenty predefined semantic relations.

Even with efforts of lots of people (about 14,000 people contributed to ConceptNet) in a long time (both WordNet and Cyc started almost twenty years ago), building a comprehensive knowledge base is still remote. Unstructured or general texts are still too complex for current text mining techniques. That is why a lot of research focuses only on constrained text, which is either format constrained (such as tables) or content constrained (such as extracting only location information). In the rest of this section we will discuss some techniques on named entity extraction and table analysis, which are related to our case study.

Named Entity Extraction

Named entity detection and extraction techniques try to locate and extract the entity names (such as of company, people, locations (Li, et. al., 2003), biological terms (Goutte, et. al., 2002), etc.), dates (Mckay & Cunningham, 2001), monetary amounts, references (Agichtein & Ganti, 2004) and other similar entities in unstructured text. In early systems usually a domain-specific dictionary and a pattern/rule base are built manually and tuned for a particular corpus. Extraction quality depends on the quality of these external dictionaries and bases, sufficiency of training and consistency of documents within the corpus. Recently more systems utilize context information to deal better with inconsistency among documents, which results in a more robust system. In (Cohen & Sarawagi, 2004) a semi-Markov model is proposed to make better use of external dictionaries. In (McCallum, Freitag, & Pereira, 2000) a maximum entropy Markov model is introduced to segment FAQ's. Maximum entropy (ME) is also used in (Borthwick, et. al., 1998) to combine diverse knowledge sources. Both hidden Markov model (HMM) and ME can generate statistical models of words and simple word features. Document (not the whole corpus) specific rules are learned for named entities extraction to keep more knowledge of original documents (Callan & Mitamura, 2002).

Named entity extraction focuses on extracting simple terms, hopefully to get some insights for development of general NLP techniques. However, a named entity often has semantic relations with other parts of text, and focusing on only named entities ignores these semantic connections. Instead we choose text with constrained structures, such as tables for our case study. Table is semantically complete and usually rich in information.

Table Analysis

Tables are widely used in documents, and are self-contained in semantics and structure. Unstructured text, semi-structured text (such as HTML, LATEX), structured text (such as XML), all utilize table to represent information with repeated patterns.

There exists a lot of work in table analysis, and usually they can be divided into (Zanibbi, Blostein, & Cordy, 2004):

- table detection
- table modeling
- table structure analysis
- table information extraction

After a table is detected, physical and logical structures of tables are studied. Data structures and operations are defined for more complex table processing, such as table regeneration, transformation and inferences (Wang & Wood, 1998). Then tables are decomposed with a table model, such as constraint-based table structure derivation (Hurst, 2001), graph theory based system (Amano & Asada, 2003), extraction using conditional random fields (Pinto, et. al., 2003). Even after table structure analysis, the task of table information extraction is still non-trivial. Table 2 shows that semantic information has to be considered, which may result in changes of original table model based on structure information, such as splitting or merging cells.

Recently due to the popularity of web pages, detection and analysis of tables in HTML documents get a lot of attention (Wang & Hu, 2002; Chen, Tsai, & Tsai, 2000). HTML provides table tags which often help detect and segment tables, but offers little help on semantic analysis. And due to inconsistent quality of web pages, erroneous tags become noise and require additional processing.

Most of above methods are developed for table analysis only. Instead, our work is primarily concerned with broader application of SenseNet in text mining, and entity extraction from tables is used as an application in this context. Consequently it is meaningless to compare our experimental results to those obtained by these methods designed just for table analysis, and often just for tables with specific structures and in a narrow domain. With information extraction from tables as a case study, we want to show SenseNet as a methodological study which can be applied more broadly. Additionally performing a fair comparison of our work with other entity extraction techniques is not straightforward due to the difficulty of obtaining the same set of documents and knowledge base used in their experiments and determining the effects of preprocessing performed in lots of those techniques.

SENSENET: A KNOWLEDGE REPRESENTATION MODEL

We divide the natural language understanding process into three phases:

- learning phase
- thinking phase
- testing phase

Learning or knowledge acquisition to set up a general purpose knowledge base requires large amounts of resources and time as shown by Cyc, WordNet and CommonSense projects. For SenseNet we could reuse the knowledge bases built by WordNet, but need to build semantic connections among word senses. In our case study, SenseNet is effectively applied to a specific domain, tables in financial documents. Even with this small domain, the amount of knowledge required is large. Difficulty of learning is a common and severe problem to any existing knowledge bases. Whether there exists an automatic learning method which can build a high quality, general-purpose, practical knowledge base is still an open question.

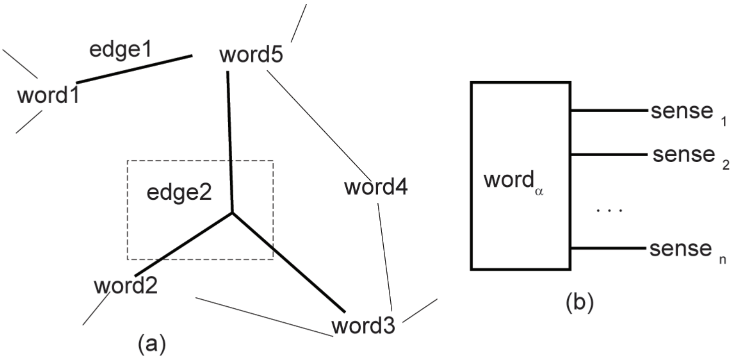
SenseNet model

Lexicon is the knowledge of words, which includes a large amount of "character string to real entity" mappings. Memorization of these mappings is difficult for human beings. It explains why in many natural languages a word often represents multiple meanings. In computational linguistics a meaning of a word is called a sense. From the view of semantics a sense is a better choice for a knowledge base than a word because a sense encodes a single and clear meaning. Our KR model, SenseNet, uses a sense as the basic semantic unit.

An instance of SenseNet is shown in Figure 2 (a). Each node represents a word. A node has multiple attributes representing the senses of a word, and each sense represents a single unambiguous entity (meaning). *Entity* is defined as "something that has independent, separate, or self-contained existence and objective or conceptual reality" by Webster dictionary. A word $word_\alpha$ is defined as the set of all its senses $\{sense_i\}$, which is shown in the Figure 2 (b), where i = 1, …, n.

A *simple edge* connects two semantically related words, for example, edge1 in Figure 2. As shown in Figure 3, a simple edge represents

Figure 2. (a) An instance of SenseNet (b) A node of SenseNet represents a word



the semantic relationship between word$_\alpha$ and word$_\beta$, that is, the probability of word$_\alpha$ taking sense i and word$_\beta$ taking sense j at the same time. A simple edge connecting word$_\alpha$ and word$_\beta$ is defined as a probability matrix:

$R_{n \times m}$ = P{word$_\alpha$ = sense$_i$, word$_\beta$ = sense$_j$}, i = 1, …, n; j = 1, …, m

R is a reflective matrix, that is, the probability of word$_\alpha$ taking the sense$_i$ if word$_\beta$ takes the sense$_j$ is equal to the probability of word$_\beta$ taking sense$_j$ and word$_\alpha$ takes sense$_i$.

A *complex edge* connects more than two words (for example, edge2 in Figure 2 connects three words, word$_2$, word$_3$, and word$_5$), which means that these words are semantically related together to express combined or more specific information. For example, to correctly analyze "give Tom a book", "give", "Tom", and "book" need to be processed together to capture the complete information. A complex edge is formally defined as:

$R_{Nw\alpha \times Nw\beta \times \ldots \times Nw\gamma}$ = P{word$_\alpha$ = sense$_i$, word$_\beta$ = sense$_j$, …, word$_\gamma$ = sense$_k$}

where sense$_i$ is a sense of word$_\alpha$, $1 \le i \le N_{w\alpha}$, $N_{w\alpha}$ is the total number of senses of word$_\alpha$; sense$_j$ is a sense of word$_\beta$, $1 \le j \le N_{w\beta}$, $N_{w\beta}$ is the total number of senses of word$_\beta$; sense$_k$ is a sense of word$_\gamma$, $1 \le k \le N_{w\gamma}$, $N_{w\gamma}$ is the total number of senses of word$_\gamma$.

A complex edge that connects m nodes is called an *m-edge*, hence a simple edge is also a 2-edge. Of course, different edges will contain different probabilities reflecting different strength among connected words.

Figure 3. An edge of SenseNet

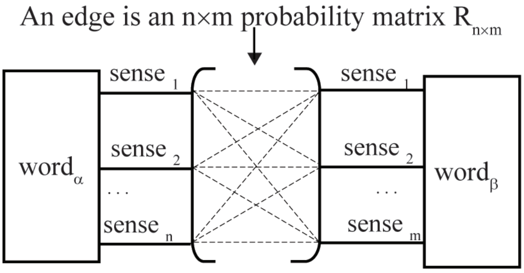
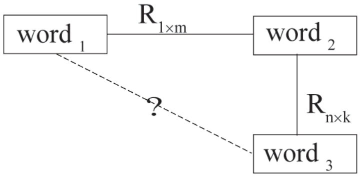


Figure 4. Implication process



Confidence

Most machine learning algorithms discard duplicate samples during training as no new information can be gained. However, the number of these identical samples indicates how often a sample occurs and how many users agree upon them. During human learning process, duplicate samples do not give new information, but will build our confidence on the indicated information. Similarly in SenseNet we use the number of identical samples as *confidence* for that sample. We define three types of confidence: sense confidence, connection confidence and global confidence.

Suppose a word w has n senses, for each sense there exists a sense confidence. A sense confidence represents the frequency that this sense is encountered during training and is normalized to a value between 0 and 1. A connection confidence is defined on a connection between two senses. Similarly, it represents the frequency of this connection is encountered during training and is also normalized to a value between 0 and 1. Global confidence shows our overall confidence of the current SenseNet, and it serves as $C_{threshold}$ in our inference algorithm discussed in Section 4.2. Global confidence is statistically derived from sense and connection confidence existed in a SenseNet, for example, it can be the average value, minimum, or maximum of all existing confidence. As shown in the inference algorithm (Section 4.2), if global confidence takes the minimum value, a great number of low-confidence senses will be activated, which mimics an over-confident human being.

Confidence can also be affected by the source of samples. For example, we may be very confident with word definitions in a dictionary. We thus assign a high confidence to these trusted sources directly. By this way training is shortened because the closer the confidence is to 1, the less learning is required. Just like a human being, if he is confident with his knowledge on a topic, he will not spend much time learning it.

Implication Operation

Training is expensive for most machine learning algorithms. To make the best use of training efforts we apply implication operation to generate new edges and expand the newly built SenseNet. We denote this phase as thinking phase.

Suppose that two edges are learned (Figure 4). Then through implication operation we try to determine whether an edge (semantic relationship) exists between $word_1$ and $word_3$. Implication operation is defined as:

$$R_{1\times k} = R_{1\times m} \times R_{m\times k}$$

where $R_{1\times m}$ is the probability matrix between $word_1$ and $word_2$, $R_{m\times k}$ is the probability matrix between $word_2$ and $word_3$, and $R_{1\times k}$ is the calculated probability matrix between $word_1$ and $word_3$. $word_1$ and $word_3$ are not semantically related if all values in $R_{1\times k}$ are zero. Otherwise, a new edge is inserted into the SenseNet between $word_1$ and $word_3$. It is possible that there exist multiple routes connecting $word_1$ and $word_3$. In this case first we will generate multiple temporary edges from these routes, then these temporary edges are averaged to generate the new edge between two words.

The confidence of the newly generated edge is the multiplication of two original edge confidence. Because confidence values have been normalized between 0 and 1, the calculated confidence is smaller than either of the original values. This process exactly simulates the learning process of human beings, as we usually have lower confidence with indirect knowledge generated by reasoning than directly taught knowledge.

Combination Operation

After multiple simple edges connecting the same set of nodes are generated, we can combine them into a complex edge. The combination of two simple edges is defined as:

$$R_{l,\,k,\,x} = R_{l,\,k,\,1} \times R_{l,\,k,\,x} \times \theta$$

where $R_{l,\,k}$ is rewritten to $R_{l,\,k,\,1}$, $R_{k,\,x}$ is rewritten as $R_{1,k,x}$, θ is within [0,1], θ shows the decreasing confidence. As the number of nodes involved becomes large, the cost of combination operation will also go up. But the new relation matrix for combined edge is usually very sparse, and storage and processing techniques of sparse matrix can be very helpful in this case. Also some hybrid techniques can be used, such a look-up table or hash table. Combination of more than two edges can be performed in a similar way.

In summary, both implication and combination operations will generate new knowledge which may be unique to a specific SenseNet, fortify the learning capabilities and reduce the training cost. In SenseNet both edges and nodes are learned and updated locally and flexibly. Therefore, like human intelligence, SenseNet is robust in dealing with inconsistent and incomplete data.

Disambiguation with SenseNet

According to SenseNet ambiguity arises when there is more than one way to activate the senses or edges. The following example shows how to use SenseNet to analyze word sense ambiguity. This process is formalized in section 4.2.

Example 1: A gambler lost his lot in the parking lot.

Webster dictionary defines "lot" as:

- an object used as a counter in determining a question by chance;
- a portion of land;
- a considerable quantity or extent;
- …

Which senses of "lot" should be activated? This problem is called word sense disambiguation in natural language processing. Because of the edge between "gambler" and "lot", "an object used as a counter in determining a question by chance" is activated for the first "lot", and "a portion of land" for the second "lot" due to its relation to "parking" (shown in Figure 5).

Another form of ambiguity lies in the syntactic structure of the sentence or fragment of language. In the next example it is not clear whether the adjective "small" applies to both dogs and cats or just to dogs.

Example 2: small dogs and cats

As shown in Figure 6, SenseNet has two options to activate edges, which leads to ambiguity.

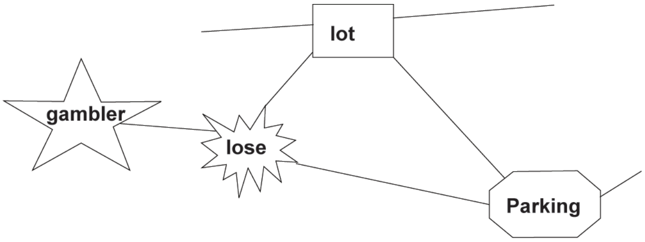
The sentence below shows an example of implication ambiguity.

Example 3: The chicken is ready to eat.

For this sentence the ambiguity comes from whether to activate another node as shown in Figure 7. Although the node "people" does not appear in the sentence, but since implication or omission is very common in communication, we may assume "people" is omitted due to simplicity. Again there are two options to activate the SenseNet, which leads to ambiguity.

Basically ambiguity arises when there are two or more ways to activate the senses, nodes or edges in a SenseNet. These examples show that flexibility and ambiguity of a natural language come from the same source. To avoid ambiguity more constraints are needed for only one activation.

Figure 5. Sense disambiguation for "lot"



NATURAL LANGUAGE UNDERSTANDING WITH SENSENET

A Hidden Markov Model (HMM) is a discrete-time finite-state automation with stochastic state transition and symbol emission (Durbin, et. al., 1998). Recently HMM is gaining popularity in text mining as researchers pay more attention to relations and context of entities (Seymore, McCallum, & Rosenfeld, 1999). HMM has been widely used for segmentation (Teahan, 2000), text classification (Hughes, Guttorp, & Charles, 1999), and entity extraction (Cohen & Sarawagi, 2004). For details about HMM, please refer to (Rabiner, 1989).

Formalizing Natural Language Understanding with a Hidden Markov Model

In SenseNet, the natural language understanding process is the process of selecting appropriate senses for each word in the text. To understand a document, a human being tries to determine meanings (senses) of words, which is an analysis and reasoning process. We formalize this process with a HMM using SenseNet as the knowledge base. Suppose there are M states in the HMM. The state at time t is s_t, where t = 0, 1, 2, …, M is the time index. The initial state s_0 is an empty set. The state s_t consists of the senses of all processed word set W_t. At time t+1, we will determine the sense of next unprocessed word w_{t+1} that has connections (edges in SenseNet) with W_t. Which sense of w_{t+1} will be activated is decided by strength (probability

Figure 6. Left side is a fragment of SenseNet. Right side shows two options to activate edges.

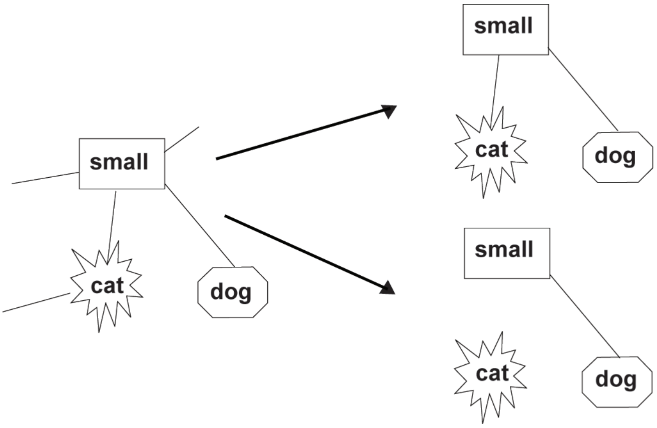
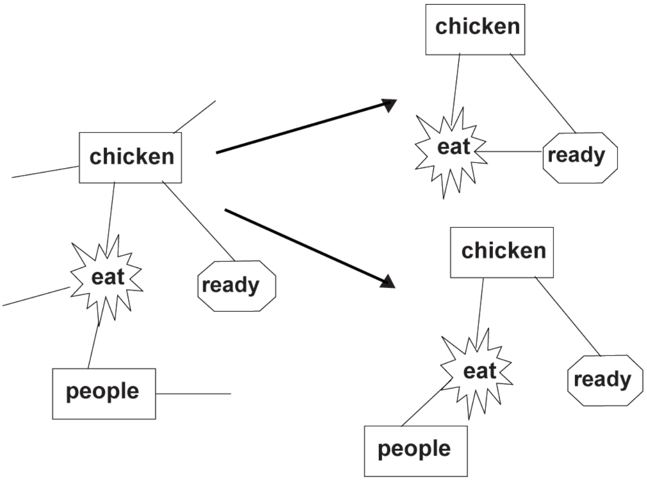


Figure 7. Whether to activate another node gives ambiguity. Left side is a fragment of SenseNet, and right side only keeps activated edges and nodes



and confidence) of edges between w_{t+1} and W_t in SenseNet. The transition from s_t to s_{t+1} is given by the conditional probability $P(s_{t+1}|W_t)$, which is specified by a state transition matrix A. Elements of A are defined as:

$$a_{ij} = P(s_{t+1} = s_t \text{ U } w_{t+1}^{j} \mid W_t = W_t^{i})$$

where j is the jth sense of word w_{t+1}, and W_t^i denotes the ith combination of senses of the words in W_t. Notice that $\sum_{ij} a_{ij} = 1$.

If probability is the only measure in determining word senses, we simply choose the w_{t+1}^{j} that has the highest probability. However, as demonstrated by human natural language understanding process, probability itself is not sufficient, thus confidence is desired to measure how confident we are with our decisions. For example, the transition with highest probability is not trustworthy if it has a very low confidence. This is guarded by the $C_{threshold}$ in our inference algorithm in section 4.2. HMM has so-called "zero-frequency problem" (Witten & Bell, 1991) if transitions of zero probability (no training samples) are activated. SenseNet solves this problem by assigning a small value to every transition as its initial probability.

In SenseNet, suppose there is a node w_i, confidence for its jth sense is denoted as c_{ij}. Suppose there are two related nodes, w_i and w_m, the confidence of probability connecting their jth and nth sense is denoted by $c_{ij,mn}$. We define that the confidence for the overall SenseNet C as average of all sense confidence and relation confidence. We use C as $C_{threshold}$ in our inference algorithm during testing phase.

Inference Algorithm for Natural Language Understanding

The inference problem of a regular HMM is to find the state with highest probability, which is efficiently solved by Viterbi algorithm (Viterbi, 1967). However, in SenseNet the goal is to find a state set S with high probability and confidence for a given document, which consists of the word sequence $W = w_1, w_2, \ldots, w_n$. Thus, the inference algorithm returns all states that satisfy:

$$S = \{ s_i \mid P(s_i|W) > P_{threshold}, C(s_i|W) > C_{threshold} \}$$

where $P_{threshold}$ and $C_{threshold}$ are the minimum requirements for probability and confidence. S is generated from the line 21 to 26. If S is empty, either SenseNet does not have enough knowledge or the document is semantically wrong; if S has one state, SenseNet understands the document unambiguously; if S has multiple states, there exist multiple ways to understand the document, which results in ambiguity. Ambiguity is very common in a natural language. With SenseNet we can successfully detect and analyze ambiguity. Here is the SenseNet inference algorithm for sense disambiguation.

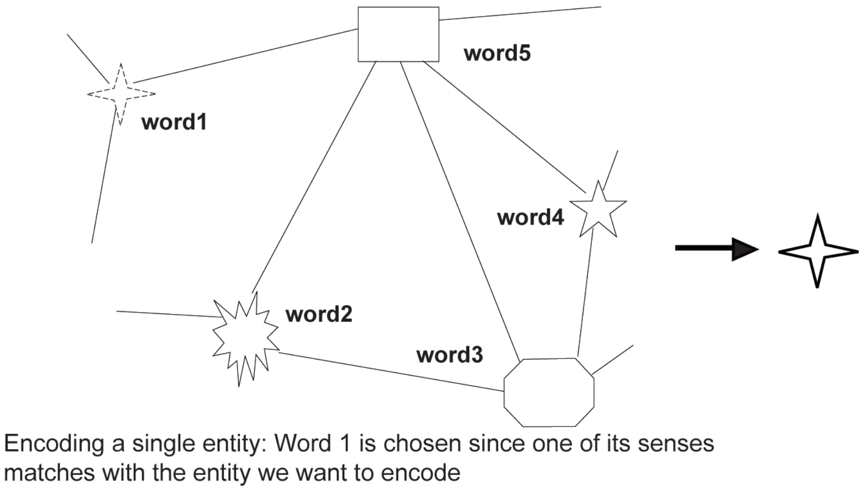
```
Inference ($W = w_1, w_2, \ldots, w_n$) {

1.  $S = \Omega$;
2.  put a word with the highest confident sense into $W^0$, choose the first one if more than one word have the same sense confidence;
3.  for each sense i of word(s) in $W^0$ {
    4.  $TBD_i = W - W^0$;
    5.  $S_i = W^0$;
    6.  for each state $s_{ik}$ in $S_i$ {
        7.  $P_{ik} = P(s_{ik})$;
        8.  $C_{ik} = C(s_{ik})$;
        9.  $TBD_{ik} = TBD_i$;
        10. while $TBD_{ik}$ is not empty {
            11. choose any words in $TBD_{ik}$ that have edges to words in $s_{ik}$, add them to $s_{ik}$, these newly added words are denoted as W', activate the senses with the highest probability;
            12. $TBD_{ik} = TBD_{ik} - W'$;
            13. $P_{ik} = P_{ik} \times P(newly\_added\_edges)$;
            14. $C_{ik} = C_{ik} \times C(newly\_added\_edges) \times C(newly\_added\_senses)$;
            15. if $C_{ik} < C_{threshold}$ or $P_{ik} < P_{threshold}$
            16. remove $s_{ik}$ from $S_i$, go to 6;
        17. }; // end of $TBD_{ik}$ loop
    18. }; // end of $S_i$ loop
    19. $S = S \cup S_i$;
20. }; // end of $W^0$ loop
21. if S is empty
    22. output "failure";
23. else if there is only one state in S
    24. output this state as result;
25. else
    26. output all states, their probabilities and confidences;
27. }
```

The inference algorithm simulates how a human being interprets documents. It starts with a word that owns a sense with the highest confidence (line 1 - 2). If there exist multiple such words, we choose the first one occurring in the document. Then the algorithm performs a breath-first searching of all possible paths with probability and confidence above given thresholds and save them into S (line 3 - 20). If a word in S^0 has multiple senses, all of them are enumerated by the loop starting at line 3. Within the loop TBD_i (TBD means "to be determined") saves all unprocessed words; S_i saves all partial state sequences found so far for the ith sense. Then the algorithm tries to complete each partial state sequence by activating the related senses in SenseNet (line 11). During the process, the probability and confidence for each state sequence are updated with newly added edges and senses. If either probability or confidence falls below its threshold, this state sequence is discarded (line 16). P(newly_added_edges) in line 13 is the product of probabilities of all newly added edges; C(newly_added_edges) in line 14 is the product of confidences of all newly added edges, and C(newly_added_senses) is the product of confidences of all newly added senses. Line 19 saves all qualified state sequences into S. As more words in W are processed, P_{ik} and C_{ik} become lower, which precisely mimics the process of human natural language understanding. When a human being reads a long and hard article, he feels more and more confused and less and less confident.

ANALYZING COMMUNICATION PROCESS USING SENSENET

A natural language is a very common communication tool. There are two phases in a communication process, encoding phase and

Figure 8. Encoding process, dash-lined shape shows the original entity to be described, solid-lined shape shows the word chosen to represent it. Mismatch is possible due to language or user constraints



Encoding a single entity: Word 1 is chosen since one of its senses matches with the entity we want to encode

decoding phase. Encoding generates texts from SenseNet, and decoding converts texts to a multiple dimensional model with help of SenseNet.

Encoding and Decoding at the Single Entity Level

Let's look at how a single entity is processed first. In the encoding phase, the language generator (either a human being or a machine) searches a vocabulary base for a word to represent the entity. Multiple matches are possible, and mismatch often exists due to the constraints of the language or insufficient learning of the language as shown in Figure 8.

In the decoding phase a receiver is able to figure out the meaning or represented entity from a single word only if the word has only one sense as shown in Figure 9.

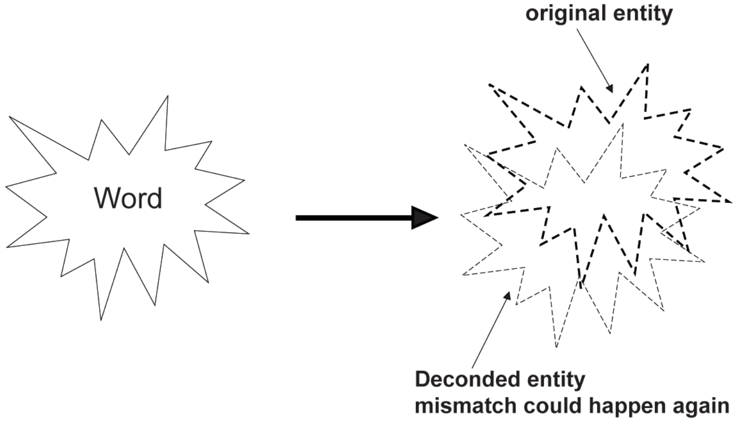
Encoding or decoding a single entity is somewhat pointless since usually more constraints are necessary to reach a decision.

Encoding and Decoding at the Scenario Level

When a scenario is to be described, usually there involve multiple entities. After choosing nodes for every entity, the language generator will organize these words together into texts as shown in Figure 10. In this organization or encoding phase, a multiple dimensional model is transformed to one dimensional text based on heuristics and syntax. During this process prepositions are used to encode time or space information, and conjunctions are used to further specify or constrain the relations among nodes.

In the decoding phase, when the receiver tries to find out what entity each word describes by performing the sense determination process described previously, and convert the one dimensional text back to multiple dimensional information modeled in SenseNet. The whole process is shown in Figure 11. It includes activation of senses, nodes and edges. And it is clear that the relations among words provide the only way for us to determine the word senses and identify the original entities.

Figure 9. Decoding process, solid-lined shape shows the word a receiver gets, and mismatch can happen

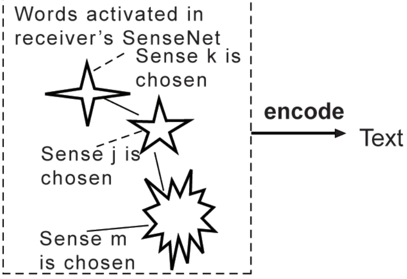


In the scenario case, not all entities play the same roles or of the same importance. In efficient communications, such as a well-written article, there usually exist a few "core" entities which connect to lots of entities. Naturally these "core" entities can be used as keywords or for text summarization. Efficient communication is also affected by careful encoding, decoding capabilities and common knowledge shared by encoders and decoders.

A CASE STUDY

We used a corpus of public company proxy filings retrieved from the online repository of the United States Securities and Exchange Commission (SEC). SEC names these documents as DEF 14A. Every DEF 14A contains one executive compensation table (e.g., Table 1 and Table 2). There exist a wide range of structural differences among these tables, such as different number of lines or columns for each executive entry, incomplete data. As shown in Table 1, without semantic information we cannot understand that this table describes compensation of two executives for three years. An example of ambiguity is shown in Table 2, "Jr" could be a suffix for "Ed J. Rocha", or a prefix for "CFO". Utilization of mere structural information results in a "brittle" system.

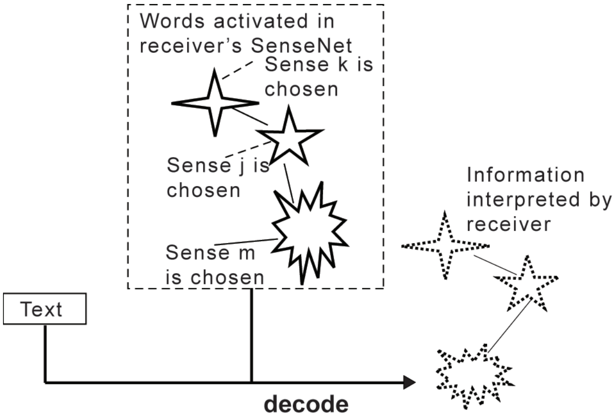
Figure 10. Encoding at the scenario level



We built an Executive Compensation Retrieval System (ECRS) to extract the data fields from these tables and save them in a database. ECRS includes,

- a web crawler to download the latest DEF 14A regularly.
- a knowledge base generated from a list of personal names from the U.S. Census Bureau and a list of titles of company executives. According to the Census Bureau, this name list contains approximately 90 percent of all of the first and last names in use in the U.S. The list was partitioned by first and last name and the total number of entrees is 91,933. Each first or last name

Figure 11. Decoding at the scenario level


Words activated in receiver's SenseNet
Sense k is chosen
Sense j is chosen
Sense m is chosen
Text
decode
Information interpreted by receiver

will be a node in SenseNet, and there exist one edge between each pair of first name and last name. For the company executive title list, titles were manually extracted from about 25 randomly picked financial documents. Example titles include Chief Executive Officer, CFO, Chairman, Chief, and CIO etc. We converted this list into SenseNet with each word as a node, and there are edges for words appearing in one title. We found that some words appear in both the name and title list, such as "president", "chairman". And these words have two senses and require disambiguation. Since the names and titles come from trusted sources, we assign all confidence values as 1.

- an extraction module, which locates executive compensation tables and extracts executive names, titles, salary, bonus, stock options and other data fields.
- a database that saves all the extracted information.

The experiment was conducted using randomly picked Standard and Poor's 500 companies from different industries based on Global Industry Classification Standard: 1. Automobile; 2. Bank; 3. Commercial Supply and Service; 4. Energy; 5. Food Beverage and Tobacco; 6. Health Care; 7. Insurance; 8. Pharmaceutical and Biotechnology; 9. Real Estate; 10. Software and Service; 11. Transportation. Since the only way to validate the results is by manual checking, a large-scale experiment is not feasible. Instead, we try to diversify the DEF 14A used in the experiment. At least one company of each industry was selected, and the total number of tested companies is 19. Depending on availability one to three years' reports were retrieved for each company. Total number of compensation records in these documents is 184. 149 of them are successfully extracted as shown in Table 3.

CONCLUSION AND FUTURE WORK

This paper presents a new Knowledge Representation model called SenseNet at the lexical level. We formalize SenseNet model with HMM. SenseNet models some important aspects of human reasoning in natural language understanding, can dissolve ambiguity, and simulate human communication process. We evaluate the SenseNet model by an application in table extraction. To achieve human-level intelligence there are still many open problems, e.g.,

- How to build a high-quality commonsense knowledge base automatically?

Table 1. A segment of a DEF 14A Form

Name	Year	Salary	Bonus	Other compensation
Edwin M. Crawford Chairman of Board and Chief Executive Officer	2003	1500000	127456	...
	2002		103203	...
	2001	1294231	207299	...
A.D. Frazier, Jr President and Chief Operating Officer	2003	1000000	450000	...
	2002	392308	418167	...
	2001	N/A	N/A	...
...				

Table 2. Another sample table from a SEC DEF 14A Form

Name and Position	Year	Salary	Other compensation
CAPITAL CORP OF THE WEST Thomas T. Hawker President/CEO	2000	181,538	...
	1999	173,115	...
	1998	170,219	...
COUNTY BANK Ed J. Rocha Jr. CFO	2000	118,750	...
	1999	104,167	...
	1998	N/A	…
...			

Table 3. Information extraction results

Industry	Number of years	Number of records	Extracted records
1	2	10	5
2	2	18	15
3	3	27	25
4	1	3	2
5	3	15	13
6	2	40	34
7	3	18	13
8	3	12	8
9	1	3	3
10	2	20	15
11	2	18	16

- How to build knowledge at a higher level of granularity than lexicon (such as frame)?

REFERENCES

Agichtein, E., & Ganti, V. (2004). Mining Reference Tables for Automatic Text Segmentation. *Tenth ACM International Conference on Knowledge Discovery and Data Mining*, Seattle, WA

Amano, A., & Asada, N. (2003, August). Graph Grammar Based Analysis System of Complex Table Form Document. *Seventh International Conference on Document Analysis and Recognition Volume II*, Edinburgh, Scotland.

Berwick, R., Knight, T., Shrobe, H., Sussman, G., Ullman, S., Winston, P., & Yip, K. (2004). *The Human Intelligence Enterprise*. Retrieved from http://genesis.csail.mit.edu/HIE/white.html.

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

Callan, J., & Mitamura, (2002). T. Knowledge-based extraction of named entities. *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (pp. 532-537). McLean, VA.

Chen, H., Tsai, S., & Tsai, J. (2000, August). Mining Tables from Large Scale HTML Texts. *18th International Conference on Computational Linguistics,* Germany.

Cohen, W., & Sarawagi, S. (2004). Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods. *Tenth ACM International Conference on Knowledge Discovery and Data Mining,* Seattle, WA.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge UK: Cambridge University Press.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradfords Books, ISBN 0-262-06197-X.

Goutte, C., Déjean, H., Gaussier, E., Cancedda, N., & Renders, J. (2002). Combining labelled and unlabelled data: a case study on Fisher kernels and transductive inference for biological entity recognition. *Sixth Conference on Natural Language Learning*, Taipei, Taiwan.

Hughes, J., Guttorp, P., & Charles, S. (1999). A non-homogeneous hidden Markov model for precipitation occurrence. *Applied Statistics*, *48*, 15–30. doi:10.1111/1467-9876.00136

Hurst, M. (2001, September). Layout and language: Challenges for table understanding on the web. Proceeding of International Workshop on Web Document Analysis (pp. 27-30), Seattle, USA.

Li, H., Srihari, R., Niu, C., & Li, W. (2003). Cymfony A hybrid approach to geographical references in information extraction. *Human Language Technology conference: North American chapter of the Association for Computational Linguistics annual meeting*, Edmonton, Canada.

Liu, H., & Singh, P. (2004). Commonsense reasoning in and over natural language. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES-2004)*.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proceedings of 17th International Conf. on Machine Learning*, San Francisco, CA.

Mckay, D., & Cunningham, S. (2001, April). Mining dates from historical documents. *The Fourth New Zealand Computer Science Research Students Conference,* New Zealand.

Pinto, D., McCallum, A., Wei, X., & Croft, W. (2003). Table extraction using conditional random fields. *Proceedings of the ACM SIGIR.*

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. doi:10.1109/5.18626

Seymore, K., McCallum, A., & Rosenfeld, R. (1999). Learning hidden Markov model structure for information extraction. *AAAI Workshop on Machine Learning for Information Extraction.*

Teahan, W., Wen, Y., McNab, R., & Witten, I. (2000, September). A compression-based algorithm for Chinese word segmentation. *Computational Linguistics, 26*(3, 375–393.

Viterbi, A. (1967, April). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, *IT-13*(2), 260–269. doi:10.1109/TIT.1967.1054010

Wang, X., & Wood, D. (1998). A Conceptual Model for Tables, Principles of Digital Document Processing PODDP '98.In E. Munson, C. Nicholas, & D. Wood (Eds.), *Springer-Verlag Lecture Notes in Computer Science 1481*(1998), 10-23.

Wang, Y. (2007, July). The OAR Model of Neural Informatics for Internal Knowledge Representation in the Brain. [Hershey, PA, USA: IGI Publishing.]. *International Journal of Cognitive Informatics and Natural Intelligence*, *1*(3), 64–75.

Wang, Y. (2008). On Visual Semantic Algebra (VSA) and the cognitive process of pattern recognition. *7th IEEE International Conference on Cognitive Informatics* (pp. 384-393), Stanford, CA

Wang, Y., & Hu, J. (2002, August). Detecting Tables in HTML Documents. In D. Lopresti, J. Hu, & R. Kashi (Eds.), *Document Image Analysis System V, 5th International Workshop DAS 2002*, Princeton, NJ, USA.

Wang, Y., & Wang, Y. (2006, March). Cognitive Informatics Models of the Brain. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *26*(2), 203–207. doi:10.1109/TSMCC.2006.871151

Wang, Y., & Wang, Y. (2008). The cognitive processes of consciousness and attention. *7th IEEE International Conference on Cognitive Informatics* (pp. 30-39), Stanford, CA.

Witbrock, M., Baxter, D., & Curtis, J. (2003). An Interactive Dialogue System for Knowledge Acquisition in Cyc. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.

Witten, I., & Bell, T. (1991). The zero-frequency problem: Estimating the probablitiies of novel events on adaptive text compression. *IEEE Transactions on Information Theory*, *37*(4), 1085–1094. doi:10.1109/18.87000

Zanibbi, R., R., Blostein, D., & Cordy, J. (2004, March). A Survey of Table Recognition: Models, Observations, Transformations, and Inferences. *International Journal of Document Analysis and Recognition*, *7*(1), 1–16.

Zheng, L., Luo, F., Shan, C., & Yin, W. (2008). A novel cognitive model of reading: Neuropsychology research on internal processing of the brain. *7th IEEE International Conference on Cognitive Informatics* (pp. 122-127), Stanford, CA.

Ping Chen is an associate professor of computer science and the director of Artificial Intelligence Lab at the University of Houston-Downtown. His research interests include bioinformatics, data mining, and computational semantics. Dr. Chen has received three NSF grants and published over 30 papers in major data mining, artificial intelligence, and bioinformatics conferences and journals. Dr. Chen received his BS degree on Information Science and Technology from Xi'an Jiao Tong University, MS degree on computer science from Chinese Academy of Sciences, and PhD degree on Information Technology at George Mason University.

Wei Ding received her PhD in computer science from the University of Houston in May 2008, then joined the Department of Computer Science of UMass Boston as an assistant professor in Fall 2008. Wei received her BS degree in computer science and applications from Xi'an Jiao Tong University in 1993 and her MS degree in software engineering from George Mason University in 2000. Wei is currently serving as a program committee member for the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008), International Workshop on Spatial and Spatiotemporal Data Mining (SSTDM 2008), the 17th International Conference on Software Engineering and Data Engineering, and she also served as a session chair for the 2007 IEEE International Workshop on Spatial and Spatio-temporal Data Mining in cooperation with IEEE ICDM. She is a member of the ACM and the IEEE.

Chengmin Ding is a managing consultant at IBM. Mr. Ding got his MS degree on computer science in 1998 from American University, and BS degree on computer science in 1994 from Xian Jiaotong University. Mr. Ding's current research interests include leveraging the open source UIMA framework to deliver advanced text mining/analytics solutions and evaluating activity based collaboration paradigm. His long term professional inspiration is to live in the semantic web and have free intelligent agents to handle the daily chores.