SenseNet: A Knowledge Representation Model for Computational Semantics

Ping Chen (contact author)
Department of Computer and Mathematical Sciences
University of Houston-Downtown
One Main St.
Houston, TX 77002
chenp@uhd.edu

Wei Ding
Department of Computer Science
University of Houston-Clear Lake
2700 Bay Area Blvd.
Houston, TX 77058
ding@uhcl.edu

Chengmin Ding
Institutional Shareholder Services
2099 Gaither Rd.
Rockville, MD 20850
cmding@cox.net

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 text analysis procedure. A new measurement, confidence, is introduced to facilitate the text analysis. We present a detailed case study of applying SenseNet to retrieving compensation information from company proxy filings.

KEYWORDS: Knowledge Representation, Computational Intelligence, Computational Semantics, Hidden Markov Model, Natural Language Processing, Information Retrieval, Named Entity Extraction, WordNet

SenseNet: A Knowledge Representation Model for Computational Semantics

Ping Chen
Dept. of Computer and Mathematical Sciences
Univ. of Houston-Downtown
One Main St.
Houston, TX 77002
chenp@uhd.edu

Wei Ding
Dept. of Computer Science
Univ. of Houston-Clear Lake
2700 Bay Area Blvd.
Houston, TX 77058
ding@uhcl.edu

Chengmin Ding
Inst. Shareholder Services
2099 Gaither Rd.
Rockville, MD 20850
cmding@cox.net

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 text analysis procedure. A new measurement, confidence, is introduced to facilitate the text analysis. We present a detailed case study of applying SenseNet to retrieving compensation information from company proxy filings.

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

1 Introduction

A knowledge representation (KR) technique captures the properties of real world entities and their relationships. Enormous amounts of intervened entities constitute a highly complex multiple dimensional structure. Thus a KR method needs powerful expressiveness to model such information.

Since 1960's many KR techniques have been proposed, such as semantic network, frame, scripts, induction rules etc. However, it is a daunting problem 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 processing.

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 describes a real world application on information extraction. Finally we conclude in section 6.

2 Related work

For decades, artificial intelligence (AI) researchers have recognized the importance of representing relationships among words in a commonsense knowledge base. There exist three major general-purpose knowledge bases, Cyc, WordNet, and ConceptNet.

WordNet [4] is a widely used semantic resource in computational linguistics community. It is a manually built database consisting of linked words. These words are organized into synonym sets called synsets, and each synset represents one lexical concept. Links are predefined semantic relationships among words. It has taken WordNet more than 10 years to collect 150,000 words/strings and 110,000 synsets. Fixed links are lack of flexibility and adaptiveness.

Cyc [10] emphasizes on the formalization of commonsense knowledge into a logical framework. Similar with WordNet, its knowledge base is handcrafted by knowledge engineers. In order to use Cyc, a natural language has to be transformed to a proprietary logical representation, which is complex and expensive for real world applications.

ConceptNet [6] is proposed in the Open Mind Common Sense project in MIT. Thousands of common people contributed through the Web by inputting sentences in a fill-in-the-blank fashion. Then concepts and binary-relational

assertions are extracted to form ConceptNets semantic network. Currently ConceptNet contains 1.6 million edges connecting more than 300,000 nodes. Nodes are semi-structured English fragments, interrelated by an ontology of twenty predefined semantic relations.

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

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

3 SenseNet: a knowledge representation model

3.1 The 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. 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 1 (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 1 (b), where $i = 1, \cdots, n$.

A *simple edge* connects two semantically related words, for example, $edge1$ in Figure 1. As shown in Figure 2, a simple edge represents 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

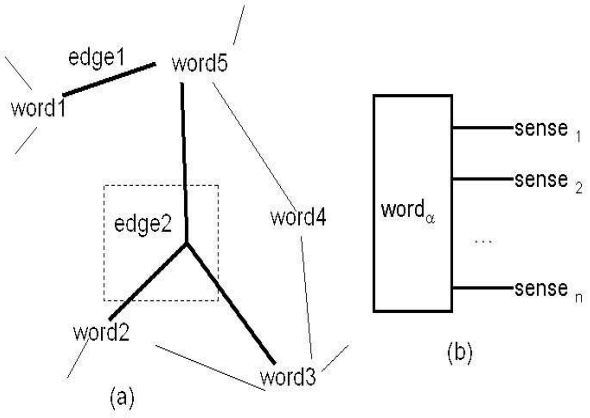

Figure 1. (a) An instance of SenseNet (b) A node of SenseNet represents a word

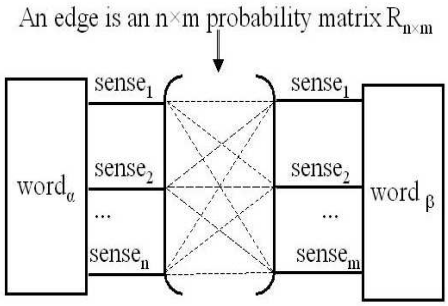

Figure 2. An edge of SenseNet

simple edge connecting $word_\alpha$ and $word_\beta$ is defined as a probability matrix:

$$R_{n\times m} = P\{word_\alpha = sense_i, word_\beta = sense_j\} \quad i = 1, \cdots, n; j = 1, \cdots, m \quad (1)$$

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 1 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:

$$R_{N_{w_\alpha}\times N_{w_\beta}\times\cdots\times N_{w_\gamma}} = P\{word_\alpha = sense_i,$$

$$word_\beta = sense_j, \quad \cdots \quad , word_\gamma = sense_k\} \quad (2)$$

where $sense_i$ is a sense of $word_\alpha$, $1 \leq i \leq N_{w_\alpha}$, N_{w_α} is the total number of senses of $word_\alpha$;

$sense_j$ is a sense of $word_\beta$, $1 \leq j \leq N_{w_\beta}$, N_{w_β} is the total number of senses of $word_\beta$;

$sense_k$ is a sense of $word_\gamma$, $1 \leq k \leq N_{w_\gamma}$, N_{w_γ} is the total number of senses of $word_\gamma$.

A complex edge that connects m nodes is called a $m - edge$, hence a simple edge is also a $2 - edge$.

3.2 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_α 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 Figure 5. 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 (Figure 5), 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.

3.3 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.

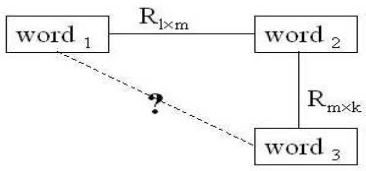

Figure 3. Implication process

Suppose that two edges are learned (Figure 3). 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:

$$R_{l\times k} = R_{l\times m} \times R_{m\times k} \quad (3)$$

where $R_{l\times m}$ 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 $R_{l\times k}$ is the calculated probability matrix between $word_1$ and $word_3$. $word_1$ and $word_3$ are not semantically related if all values in $R_{l\times k}$ 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.

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.

3.4 Disambiguation with SenseNet

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

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

Webster dictionary defines "lot" as:

1. an object used as a counter in determining a question by chance;

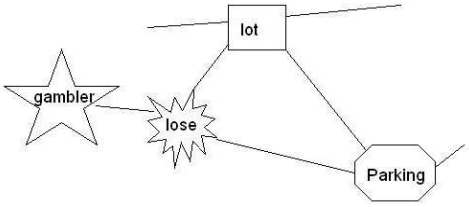

Figure 4. Sense disambiguation for "lot"

2. a portion of land;

3. a considerable quantity or extent; $\cdots$

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 4).

4 Text Analysis with SenseNet

A Hidden Markov Model (HMM) is a discrete-time finite-state automation with stochastic state transition and symbol emission [3]. Recently HMM is gaining popularity in text mining as researchers pay more attention to relations and context of entities[7]. HMM has been widely used for segmentation [8], text classification [5], and entity extraction [2]. For details about HMM, please refer to [1].

4.1 Formalizing text analysis process with a Hidden Markov Model

In SenseNet, the text analysis 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 $t = 0, 1, 2, \cdots, M$ 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 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:

$$a_{ij} = P(s_{t+1} = s_t \cup w_{t+1}^j | W_t = W_t^i) \quad (4)$$

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 $\sum_i \sum_j a_{ij} = 1$.

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 text analysis 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" [11] 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.

4.2 Inference algorithm for text analysis process

The inference problem of a regular HMM is to find the state with highest probability, which is efficiently solved by Viterbi algorithm [9]. 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 $W = w_1, w_2, \cdots, w_n$. Thus, the inference algorithm (Figure 5) returns all states that satisfy:

$$\begin{aligned} S = \{s_i | P(s_i|W) &> P_{threshold}, \\ C(s_i|W) &> C_{threshold}\} \end{aligned} \quad (5)$$

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.

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

```
Inference(W = w_1, w_2, ..., w_n) {
1  S = φ;
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 their senses with highest probability;
12             TBD_ik = TBD_ik - W';
13             P_ik = P_ik × P(newly_added_edges);
14             C_ik = C_ik × C(newly_added_edges)
               × 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 ∪ 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 }
```

Figure 5. Inference algorithm of SenseNet

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 text analysis. When a human being reads a long and hard article, he feels more and more confused and less and less confident.

5 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(table 1). 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 can not understand that this table describes compensation of two executives for three years. Utilization of mere structural information results in a "brittle" system.

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

1. a web crawler to download the latest DEF 14A regularly.

2. 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 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.

3. an extraction module, which locates executive compensation tables and extracts executive names, titles, salary, bonus, stock options and other data fields.

4. 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,

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

Table 1. A segment of a DEF 14A Form

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

Table 2. Information extraction results

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 is 184. 149 of them are successfully extracted (table 2).

6 Conclusion and future work

SenseNet models some important aspects of human reasoning in natural language processing, and has nice properties as a lexical knowledge base. However, to achieve human-level intelligence there are still many open problems that we are working on, for example,

1. Does there exist an automated method to build a high-quality commonsense knowledge base?

2. How to build knowledge at a higher level of granularity than lexicon (such as frame, script, etc.)?

References

[1] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257286. 1989.

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

[3] R. Durbin, S. Eddy, A. Krogh, G. Mitchison. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge UK, 1998.

[4] C. Fellbaum(editor), WordNet: An Electronic Lexical Database, published by Bradfords Books, ISBN 0-262-06197-X, 1998.

[5] J. Hughes, P. Guttorp, S. Charles. A nonhomogeneous hidden Markov model for precipitation occurrence. Applied Statistics, 48(1):1530, 1999.

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

[7] K. Seymore, A. McCallum, R. Rosenfeld. Learning hidden Markov model structure for information extraction. In AAAI Workshop on Machine Learning for Information Extraction, 1999.

[8] W. Teahan, Y. Wen, R. McNab, I. Witten. A compression-based algorithm for Chinese word segmentation. Computational Linguistics, 26(3):375393, September 2000.

[9] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transaction on Information Theory, IT-13(2): page 260 - 269, April 1967.

[10] M. Witbrock, D. Baxter, J. Curtis, et al. An Interactive Dialogue System for Knowledge Acquisition in Cyc. Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[11] S. Yeates, Text Augmentation: Inserting XML tags into natural language text with PPM Models and Viterbi-like search. Ph.D thesis, Computer Science Dept., Univ. of Waikato, New Zealand, 2003.