A Construction of Knowledge Graph for Semiconductor Industry Chain Based on Lattice-LSTM and PCNN Models

Charles Chen[1], *Sai-Sai Shi*[2], *Sheng-Lung Peng*[3*]

[1] *School of Business, Minnan Normal University, China*
[2] *School of Computer Science and Engineering, Minnan Normal University, China*
[3] *Department of Creation Technologies and Product Design, National Taipei University of Business, Taiwan*
charles.chen@mnnu.edu.cn, 1575102167@qq.com, slpeng@ntub.edu.tw

Abstract

This paper mainly focuses on building the knowledge graph of semiconductor industry chain. The main research contents include knowledge extraction, knowledge storage, and construction of knowledge graph in semiconductor field. The crawler technology and character recognition technology are used to obtain semiconductor industry chain information from the Internet, magazines, and institutions to establish the original data set. Then, Lattice Long Short-Term Memory (Lattice-LSTM) model is used to implement the entity extraction and recognition. The piecewise convolutional neural network (PCNN) model based on the sentence-level attention mechanism is used to extract relationships and obtain entity triples. The semiconductor dictionary library is constructed through the obtained structured data. The dictionary library and Chinese natural language toolkit HanLP are combined to annotate unstructured text data for knowledge extraction. Neo4j graph database is used to store the extracted data of semiconductor industry chain. Finally, Spring Boot and Vue technology are used to create a knowledge graph system.

Keywords: Knowledge graph, Entity recognition, Relationship extraction, Lattice-LSTM, PCNN

1 Introduction

In 2012, Google took the lead in proposing the concept of “Knowledge Graph” (KG). The knowledge graph is essentially derived from the early semantic web [1] and is a structured semantic knowledge base with a graph structure. It can more intuitively describe the human cognitive world in the form of entity relationship and facilitate people’s cognition. Intelligently processing massive information through knowledge mapping technology can form a large-scale knowledge base and support business applications. So that machines can better understand users, understand the network, understand resources, and provide users with more intelligent services. At present, the knowledge graph has been widely used in many fields such as e-commerce, finance, enterprises, public security, medicine, and justice.

In our research, we will introduce the knowledge graph technology into the semiconductor industry, build the knowledge graph of the semiconductor industry chain, form a systematic and standardized structure of huge redundant knowledge, and optimize the accuracy and efficiency of information query. This can intuitively describe all kinds of data information and their relationships in the form of graph, can infer some potential knowledge, and can expand users’ knowledge scope and understanding depth, which plays an important role for the popularization of knowledge in the semiconductor field.

The knowledge graph has large-scale knowledge data, and the relationship between the data is complex. Compared with traditional databases, graph databases have the advantages of easy modeling, high efficiency in storing and querying massive complex relational data, and are more consistent with the storage of knowledge graphs. Therefore, this paper will select the graph database Neo4j to provide users with more systematic and intelligent services through knowledge extraction, storage and visualization of entities and their relationships of multi-source semiconductor data.

The rest content and structure of the paper are as follows. Section 2 presents related works. Section 3 is the research contents and methods. Section 4 is knowledge graph construction. Section 5 is KG query system construction. Finally, Section 6 is conclusion and future work.

2 Related Works

2.1 Knowledge Graph

The earliest knowledge graph can be traced back to the 1960s. J. R. Quillian proposed the concept of semantic network, which is mainly used as an obvious axiom model of human associative memory. It is a structured way to express knowledge with a graph. Since then, the knowledge graph has officially come into people’s eyes. With the rapid development of the Internet, the semantic network has experienced the development of ontology, Web, semantic web and linked data. Until 2012, Google commercialized some concepts based on the semantic web and proposed the concept of knowledge graph. Following Facebook’s graph search, Microsoft Satori and specific knowledge bases in business, finance, life sciences and other fields have emerged. The most representative large-scale network knowledge

*Corresponding Author: Sheng-Lung Peng; E-mail: slpeng@ntub.edu.tw

DOI: 10.53106/160792642024032502013

acquisition work includes DBpedia, Freebase, KnowItAll, WikiTaxonomy and YAGO, as well as BabelNet, ConceptNet, DeepDive, NELL, Probase, Wikidata, XLore, Zhishi.me, etc.

The knowledge graph is not a single technology, but the extraction, representation, fusion, reasoning and application process of the whole network big data. Through database, data mining, natural language understanding and other technologies, unordered data can be transformed into a knowledge network with a directed graph structure, where nodes represent entities or concepts, while edges represent the relationships between entity concepts. Its key technologies can be summarized as four processes: knowledge representation and modeling, knowledge extraction and mining, knowledge storage and fusion, knowledge retrieval and reasoning.

There are two ways to build knowledge graphs: Top Down and Bottom Up. The top-down construction method refers to defining the ontology and data pattern for the knowledge graph, which is generally suitable for the construction of the domain knowledge graph. In the process of defining ontology, first start from the top-level concept, and then gradually refine it to form a well structured hierarchy. After defining the data schema, add entities to the concept one by one. This construction method usually requires the help of structured data sources such as encyclopedic knowledge. For example, FreeBase is built in this way, and most of its data is obtained from Wikipedia. On the contrary, the bottom-up approach, first summarizes and organizes entities to form the underlying concepts, and then gradually abstracts them upward to form the upper concepts. Typical representatives of knowledge graphs constructed in this way include Google's Knowledge Vault and Microsoft's Satori.

According to the research direction and method of each person, the construction process will be slightly different. Yang et al. [2] divided the knowledge graph construction process into eight parts, including sample data collection, sample data cleaning, knowledge unit selection, unit relationship construction, data standardization, sample data simplification, knowledge visualization, and result interpretation. Borner et al. [3] divided it into six steps: extracting data, defining analysis units, selecting methods, calculating similarity, building knowledge units, and analyzing results. But they all mentioned the most important parts of knowledge graph construction: data acquisition, information extraction, knowledge fusion and construction. Figure 1 shows the construction process of knowledge graph.

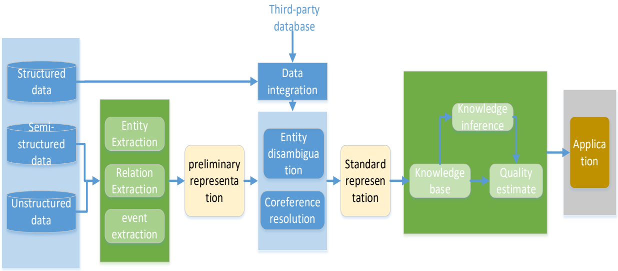


Figure 1. Knowledge graph construction process

2.2 Entity Recognition Technology

Knowledge extraction mainly refers to the technology of extracting knowledge from data of different sources and structures by using entity extraction, relationship extraction, attribute extraction, event extraction, etc. Entity extraction, *i.e.*, entity recognition, is the most critical part of knowledge extraction and the basis for building a knowledge graph. The accuracy of its methods will directly lead to the quality of the extracted knowledge entities, which will affect the performance of the application of knowledge graph functions. Especially in the vertical disciplines, it is the most difficult checkpoint in the knowledge extraction process.

Named entity recognition has always been a research hotspot in the field of natural language processing. At present, there are three common recognition methods: dictionary and rule based, statistical model based, and deep learning based.

(1) Dictionary and rule based approach

Dictionary and rule based methods are the methods used in the earliest generation of named entity recognition. Most of these methods use specific rule templates or dictionaries constructed by linguists manually based on the characteristics of data sets. Rules include keywords, location words, location words, head words, indicators, statistical information, punctuation marks, etc. A dictionary is a dictionary composed of characteristic words and an external dictionary, which refers to the existing common sense dictionary. After the rules and dictionaries are formulated, the text is usually processed by regular matching to achieve named entity recognition.

(2) Method based on statistical model

With the development of natural language processing, the method based on statistical model is proposed, and named entity recognition is regarded as a sequence annotation problem. Compared with the classification problem, the current prediction tag in the sequence labeling problem is not only related to the current input features, but also related to the previous prediction tag, that is, there is a strong interdependence between the prediction tag sequences. At present, mainstream statistical models mainly include: Hidden Markov Model (HMM) [4], Maximum Entropy (ME) [5], Maximum Entropy Markov Model (MEMM) [6], Support Vector Machine (SVM) [7], Conditional Random Fields (CRF) [8], and so on.

(3) Method based on deep learning

With the continuous development of deep learning, the research focus of named entity recognition (NER) has shifted to the Deep Neural Network (DNN), which hardly requires feature engineering and domain knowledge. Applying deep learning technology to named entity recognition has three core advantages. Firstly, NER benefits from nonlinear transformations, which generate nonlinear mappings from input to output. Compared with linear models, deep learning based models can learn complex features from data through nonlinear activation functions. Secondly, deep learning improves the efficiency of designing NER features. The NER model under the deep learning framework can generally be divided into three parts: input representation layer, context encoder, tag decoder. The function of the Input Layer is to map the discrete tokens that make up a sentence to a continuous space for later calculation. Context Encoder is used to model the semantics of words in sentences. There are two main types of tag decoders. The first is MLP + Softmax. After we get the representation of each word, we directly use a linear layer to get the score of each tag corresponding to the word. The second is Conditional Random Field (CRF), which can model the internal dependencies of tag sequences. Currently, the commonly used methods are RNN [9], BiLSTM [10], BiLSTM-CRF [11], and so on.

Some scholars mixed auxiliary information and deep learning for named entity recognition. Adding attention mechanism, graph neural network, transfer learning, far supervised learning and other popular research technologies to neural network based structures are also the mainstream research direction at present. For example, Sun et al. [12] introduced more prior knowledge by designing queries in the MRC (machine reading comprehension) framework. On the basis of enhanced word embedding, Goyal et al. [13] developed a bilingual named entity recognition system based on two-way gated loop unit and convolutional neural network (CNN).

2.3 Relationship Extraction Technology

The construction of knowledge graph not only needs to extract the required entities, but also needs to extract the relationship. Relation extraction refers to extracting the semantic relationship between two or more entities from the text. Relation extraction mainly includes rule based extraction method, supervised learning method, semi supervised learning method, and remote supervised learning method.

(1) Rule based approach

Rule based knowledge extraction is mainly to extract triplet information (entity relationship entity) from articles by manually defining some extraction rules. Under the customized rules, irrelevant phrases are removed and the remaining phrases are matched. However, the rule-based method has the same defects as the rule-based method of entity extraction. It cannot propose complex relationships, and it will consume a lot of human effort to build. It is only suitable for simple data extraction of small data.

(2) Supervised learning methods

Supervised learning uses labeled training data and traditional machine learning models or deep learning algorithms to build network models, such as C-GCN [14] and EPGNN [15]. Although supervised learning method can obtain good results by using high-quality annotation data, it requires expensive manpower and material resources to obtain high-quality annotation data.

(3) Semi supervised learning method

Semi supervised learning is a solution to the problem of obtaining a large number of high-quality annotation data. Bootstrapping learning and remote supervision methods are commonly used by using a small number of high-quality annotation data and learning through related algorithms. For the relationship extraction task, the Bootstrapping algorithm inputs a small amount of entity relationship data as the seed to find more related data with a certain relationship. However, due to searching results in large-scale data through seeds, ambiguous information may appear. The method of multi task learning can also be used to combine semi supervised relation extraction tasks with other tasks for training and learning.

(4) Remote supervised learning method

Remote supervised relation extraction is to align the corpus in large-scale unstructured text with the knowledge base, so that a large amount of training data can be obtained for model training. There are very strong assumptions or rules in the process of obtaining sample data, and there will be a lot of noise data. Text cannot clearly express the relationship between entities. Therefore, how to reduce the impact of noise data on the performance of remote supervised relation extraction is the focus of current research. Generally, reinforcement learning, confrontation learning and various attention mechanisms are used to improve data robustness.

2.4 Knowledge Storage

The traditional knowledge graph is expressed based on RDF (Resource Description Framework), which is proposed by W3C to create and process metadata. The graph database is essentially a NoSQL database. The stored graph is a group of nodes and the relationship between these nodes. Graphics store data in nodes and relationships in the form of attributes, which are key value pairs used to represent data. Different from other databases, the relationship occupies the primary position in the graph database. Based on this, it is very convenient to use the graph database to store the knowledge graph. Compared with RDF, graph database can better express real business application scenarios and pay more attention to the efficiency of data query and access, so it has more advantages in storage and query. According to the storage structure, knowledge storage can be divided into two types: knowledge storage based on table structure and knowledge storage based on graph structure, as shown in Figure 2 below. Table 1 shows some common graph databases.

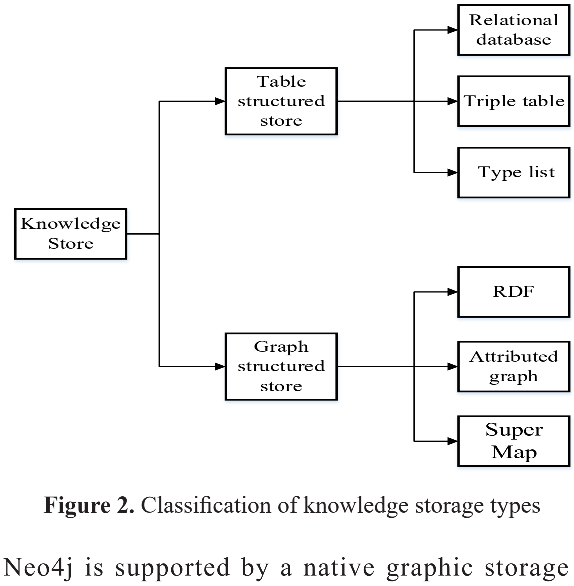
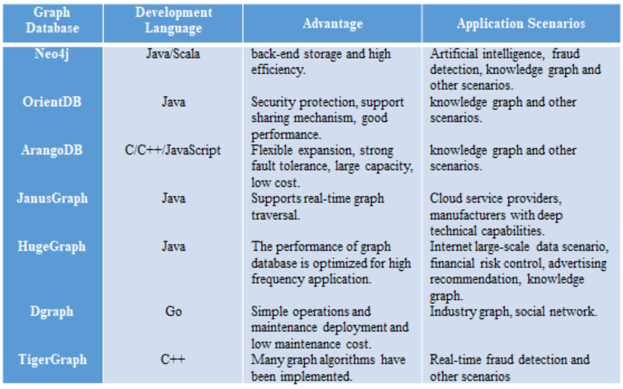


Figure 2. Classification of knowledge storage types

Neo4j is supported by a native graphic storage and processing engine, which can provide an intuitive, flexible and secure database to provide a unique and feasible storage solution. It also supports the use of CQL (Cypher) language, which is a declarative pattern matching language designed for graphic databases. Neo4j follows SQL syntax and is efficient, very simple, user-friendly and highly readable. Comparing with RDBMS, the graph model of Neo4j database is flexible, and it is good at storing and managing massive and complex semi-structured or unstructured associated data. It also can rely on Zookeeper to manage load balancing decentralized distributed storage, build a highly reliable cluster using multi replica master-slave replication, support large data sets, and constantly expand its capacity. Based on the above reasons, this paper finally selects the graph database Neo4j as the knowledge storage scheme.

Table 1. Common graph databases



Graph Database	Development Language	Advantage	Application Scenarios
Neo4j	Java/Scala	back-end storage and high efficiency.	Artificial intelligence, fraud detection, knowledge graph and other scenarios.
OrientDB	Java	Security protection, support sharing mechanism, good performance.	knowledge graph and other scenarios.
ArangoDB	C/C++/JavaScript	Flexible expansion, strong fault tolerance, large capacity, low cost.	knowledge graph and other scenarios.
JanusGraph	Java	Supports real-time graph traversal.	Cloud service providers, manufacturers with deep technical capabilities.
HugeGraph	Java	The performance of graph database is optimized for high frequency application.	Internet large-scale data scenario, financial risk control, advertising recommendation, knowledge graph.
Dgraph	Go	Simple operations and maintenance deployment and low maintenance cost.	Industry graph, social network.
TigerGraph	C++	Many graph algorithms have been implemented.	Real-time fraud detection and other scenarios

2.5 Semiconductor Related Knowledge

The process of semiconductor industry is very complex and the technical barrier is very high. According to the industrial process, it can be divided into upstream, midstream and downstream links. The upstream is mainly the preparation link, such as electronic design automation (EDA) software, IP design, semiconductor equipment, semiconductor materials, etc. The midstream mainly processes semiconductor, such as integrated circuit (IC) design, IC manufacturing, IC packaging and testing. Downstream is mainly in application fields, such as automobile, computer, manufacturing, security, communication, consumer electronics, industry, military industry, and so on. Figure 3 is the Process Flow Chart of Semiconductor.

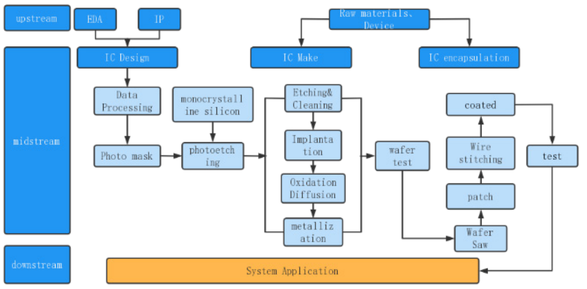


Figure 3. Process flow chart of semiconductor

2.6 Related Algorithms

2.6.1 Conditional Random Field

Conditional random fields (CRF) is a basic model in natural language processing, which is widely used in scenes such as word segmentation, entity recognition and part of speech tagging. At present, linear chain conditional random fields are mainly used to label scenes. CRF is a sequence modeling framework with all the advantages of maximum entropy Markov model, and it also solves the problem of label deviation in principle. CRF uses a single exponential model to represent the joint probability of the entire tag sequence for a given observation sequence. Therefore, the weights of different features in different states can be weighed against each other. Two graph structures of linear chain conditional random fields are shown in Figure 4.

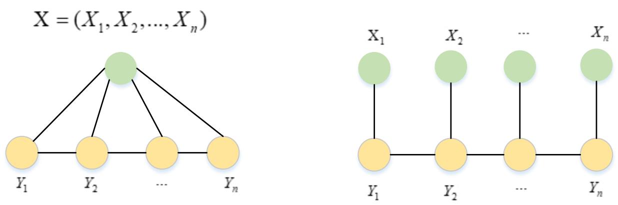


Figure 4. Two graph structures of conditional random fields

In the Figure 4, $X = (X_1, X_2, \ldots, X_n)$ is the input observation sequence, and $Y = (Y_1, Y_2, \ldots, Y_n)$ is the output annotation sequence. The linear chain CRF mainly has two important components: the transmission matrix and the transfer matrix. The emission matrix is the sum of the product of each node characteristic function and its weight; The transfer matrix is the sum of the product of each local Eigen function and its weight. The selection of feature functions in CRF is directly related to the performance of the model. The selection of features needs to consider the characteristics of the text sequence, such as domain, expression, etc. CRF is defined as follows:

$$P(y \mid x) = \frac{1}{Z}\exp\left(\sum_{i,k} \lambda_k t_k\left(y_{i-1}, y_i, x, i\right) + \sum_{i,l} u_l s_l\left(y_{i,x,i}\right)\right)$$

$$Z(x) = \sum_{y} \exp\left(\sum_{i,k} \lambda_k t_k\left(y_{i-1}, y_i, x, i\right) + \sum_{i,l} u_l s_l\left(y_{i,x,i}\right)\right),$$

where t_k is the characteristic function defined on the edge, and s_l is the characteristic function defined on the node, λ_k and μ_l is the corresponding weight value. $Z(x)$ is the normalization factor, and the sum is performed on all possible output sequences.

2.6.2 Long and Short Term Memory Network

Long Short Term Memory (LSTM) [16] is a kind of time cyclic neural network, which is specially designed to solve the long-term dependence problem of general RNN (cyclic neural network). LSTM is suitable for processing and predicting important events with very long intervals and delays in time series. The internal structure of LSTM is shown in Figure 5 below:

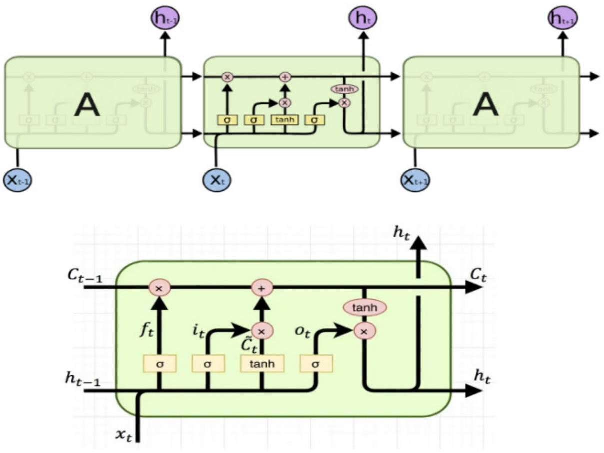


Figure 5. LSTM structure diagram (Ref. [14])

The core idea of LSTM is that the top cell state runs directly on the whole chain, similar to a conveyor belt, with only some simple linear transformations, so information can easily flow between different modules and remain unchanged. If there is only the top horizontal line, information cannot be added or deleted. It needs a structure called gates to achieve this. It is this design that gives LSTM the ability to remember long-term information. The design of the gate mainly includes three parts: input gate, forgetting gate and output gate. The formula of each part is as follows.

Input gate: $i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right)$

Forgetting gate: $f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right)$

$$\tilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right)$$

Output gate: $o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right)$

Long Term Memory: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

Short Term Memory: $h_t = o_t * \tanh\left(C_t\right)$

The first part is the forgetting gate. The forgetting layer determines which information needs to be forgotten from the cell state. The output of the upper layer and the sequence data to be inputted by this layer are used as the input, and the output is f_t through an activation function sigmoid. The second part is the input gate, which determines which new information can be stored in the cell state. The input gate consists of two parts. The first part uses the sigmoid activation function, and the output is i_t. The second part uses the tanh activation function, and the output $\tilde{C}_t$. The third part is the output gate, which determines the output value. First, we use the sigmoid activation function to get an o_t of [0,1] interval value, and then we multiply the cell state C_t by o_t after processing it with the tanh activation function, which is the output h_t of this layer.

2.6.3 Bidirectional Short and Long Term Memory Network

The basic idea of bi-directional recurrent neural network (BiRNN) is that every forward and backward input sequence passes through a cyclic neural network once. Such a bi-directional structure provides complete past and future context information for each node in the input sequence of the output layer. The bidirectional LSTM (BiLSTM) [17] replaces the ordinary RNN unit in the BiRNN with the LSTM unit. The BiLSTM neural network structure model is divided into two independent LSTMs. The input sequence is input to two LSTM neural networks in positive and reverse order respectively for feature extraction. The word vector formed by splicing the two output vectors (i.e. the extracted feature vector) is used as the final feature expression of the word. The model design concept of BiLSTM is to make the feature data obtained at time t have both past and future information. Experiments show that this neural network structure model is superior to a single LSTM structure model in the efficiency and performance of text feature extraction. The structure of BiLSTM is shown in Figure 6.

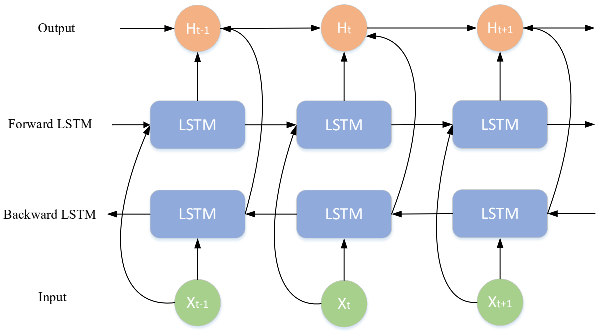


Figure 6. BiLSTM structure diagram (Ref. [15])

3 Research Contents and Methods

Knowledge extraction is the basis of building a knowledge graph, and the accuracy of extraction determines whether the knowledge graph is perfect or not. This section will focus on the establishment of the original data of the semiconductor industry chain. Firstly, entities are extracted based on Lattice-LSTM model, and then relations are extracted using piecewise convolutional neural network (PCNN) model by means of remote supervision.

3.1 Data Acquisition and Pre-processing of Semiconductor Industry

3.1.1 Data Acquisition

In this study, we use Python and BeautifulSoup4 and Requests tools to crawl HTML information of web pages for semiconductor industry chain. After investigation, we select domestic semiconductor knowledge websites with high reliability, such as electronic information industry websites (http://www.cena.com.cn), China Semiconductor Network (http://www.bandaoti.biz), and so on. For the knowledge in written magazines, we directly use manual supplementary recording to save the data in csv format. In combination with Baidu Encyclopedia, Zhihu website and other information sources, we can obtain information about semiconductor industry chain related processes, semiconductor company details, semiconductor product details and so on. Figure 7 shows the data source acquisition scope.

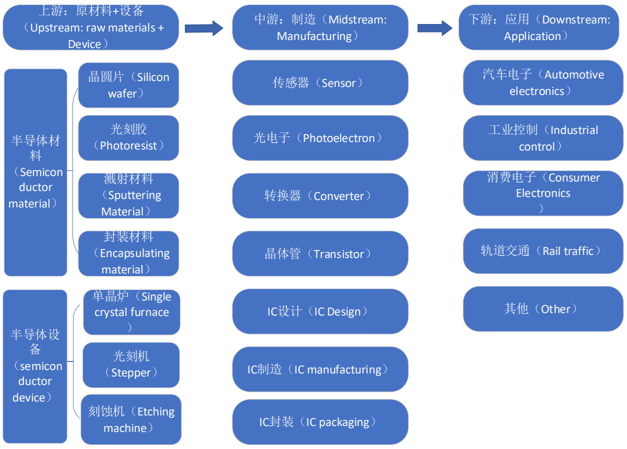

上游：原材料+设备（Upstream: raw materials + Device）
中游：制造（Midstream: Manufacturing）
下游：应用（Downstream: Application）
半导体材料（Semiconductor material）
晶圆片（Silicon wafer）
光刻胶（Photoresist）
溅射材料（Sputtering Material）
封装材料（Encapsulating material）
半导体设备（semiconductor device）
单晶炉（Single crystal furnace）
光刻机（Stepper）
刻蚀机（Etching machine）
传感器（Sensor）
光电子（Photoelectron）
转换器（Converter）
晶体管（Transistor）
IC设计（IC Design）
IC制造（IC manufacturing）
IC封装（IC packaging）
汽车电子（Automotive electronics）
工业控制（Industrial control）
消费电子（Consumer Electronics）
轨道交通（Rail traffic）
其他（Other）

Figure 7. Data source acquisition scope of semiconductor industry chain

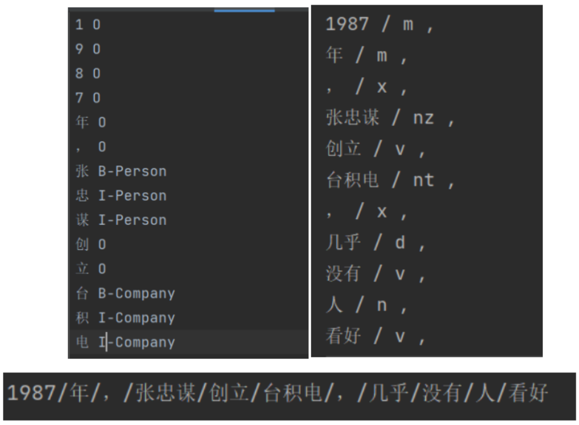
3.1.2 Pre-processing

The data in this paper is mainly obtained from websites by the crawl technology. However, the information obtained by the crawler may be partially incomplete and different. In order to reduce the interference of noise, it is necessary to preprocess the original data, mainly to remove useless labels, remove redundant punctuation marks, and some unnecessary pause words, and finally form a new sentence.

3.2 Named Entity Recognition

3.2.1 Build Features

In order to improve the performance of the prediction model, the input features of this paper mainly include character features, word boundary features, part of speech features, and entity labeling features. For the feature construction of entity annotation, this paper establishes a small entity dictionary based on the obtained structured data. The semiconductor entity dictionary library constructed in this paper mainly contains those data such as semiconductor companies, production products, semiconductor classifications, applications etc. For the convenience of construction, only 152 data of semiconductor companies, 152 executive directors, 15 products and 15 semiconductor classifications are selected. The entity label is marked by BIO (B-begin, I-inside, O-outside) sequence. It mainly aims at person name, company name, EDA software, IP module, Reticle, Photoetching, Sedimentary, Silicon, Etching, Polishing, special gas, Epitaxial, Encapsulation, Semiconductor device, IC design, IC manufacture, IC packaging and so on. B-X stands for the beginning of entity X, I-X stands for the middle or end of entity X, and O stands for not belonging to any type. For example, 台积电 (TSMC) can be labeled as 台 (B-Company) 积 (I-Company) 电 (I-Company). For part of speech and word boundary features, the corresponding labeling method of the word segmentation tool of Jieba is used. Figure 8 shows the style of data annotation.



```
1 O
9 O
8 O
7 O
年 O
， O
张 B-Person
忠 I-Person
谋 I-Person
创 O
立 O
台 B-Company
积 I-Company
电 I-Company
```

```
1987 / m ,
年 / m ,
， / x ,
张忠谋 / nz ,
创立 / v ,
台积电 / nt ,
， / x ,
几乎 / d ,
没有 / v ,
人 / n ,
看好 / v ,
```

```
1987/年/，/张忠谋/创立/台积电/，/几乎/没有/人/看好
```

Figure 8. The style of data annotation

Note. 1987 年，张忠谟创立台积电，几乎没有人看好 → In 1987, Zhang Zhongmo founded TSMC, and almost no one was optimistic about it

3.2.2 Word Embedment

Because text information and labeled information are non-computable information and cannot be inputted in the deep learning model, it is necessary to encode the information. In natural language processing, One-Hot coding is the most basic coding method. Its method is to use the N-bit status register to encode N states. Each state has its own independent register bit, and at any time, only one of the bits is valid. One Lot encoding is the representation of classification variables as binary vectors. This first requires mapping classification values to integer values. Then, each integer value is represented as a binary vector, which is zero except that the index of the integer is marked as 1. However, the vector dimension of One-Hot coding will increase with the increase of word size. If there is too much data, it will cause dimension disaster, so it is suitable for a small amount of data. In addition, this coding method is too sparse, and the efficiency of computing and storage is not high. The distributed representation of words solves this problem. By training word information, it depicts the relationship with context information. Distributed representation uses low dimensional continuous vectors to represent word information, and words with similar semantics will be similar in vector space. At present, the mainstream algorithms for word embedding include Word2Vec and Glove, among which Word2Vec is widely used. It is a software tool for training word vectors, which was disclosed by Google in 2013. Word2Vec can express a word into vector form quickly and effectively through the optimized training model according to the given corpus. Word2Vec has two model structures. One is to predict the headword based on the context information of the word, which is called CBOW model (Continuous Bag-of-Words Model). The other is to predict the context word based on the headword, which is called Skip gram model. This article uses the Skip gram model. Skip gram model structure is shown in Figure 9 below.

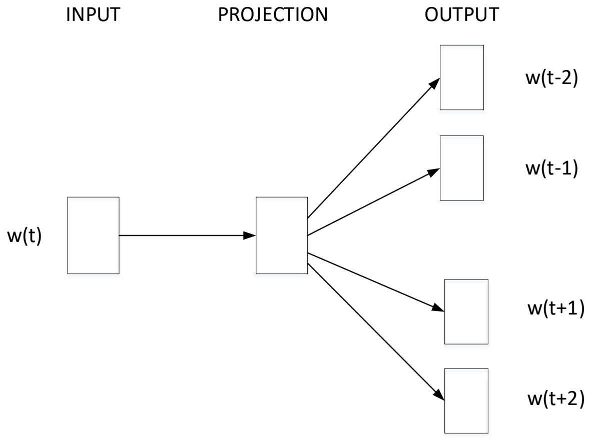


Figure 9. The structure of Skip-gram model

In the above Figure 9, w(t) is the headword, also known as the given input word. There is a hidden layer that performs the dot product between the weight matrix and the input vector w(t). The result of the dot product operation in the hidden layer is transferred to the output layer. The output layer calculates the dot product between the hidden layer output vector and the output layer weight matrix. Then the softmax activation function is used to calculate the probability of words appearing in the w(t) context at a given context position.

3.2.3 Entity Recognition Based on Lattice LSTM Model

Due to the different language structure characteristics between Chinese and English, the basic unit of Chinese is character and has not been separated, so entity recognition for Chinese is a little more difficult than that for English. It is very difficult to understand a sentence only at the word level, for example, a thief steals something secretly (小偷偷偷东西). Without word segmentation, it will be difficult to understand. Therefore, word segmentation is essential for Chinese entity recognition. However, if the sentence is segmented first, it will be more difficult to label the sequence once the marking is wrong. Lattice-LSTM [18] solved this problem. Lattice-LSTM is an improved grid structure based on BiLSTM-CRF. It forms a new vector representation for each word node and sends the word vector to the context encoder, and modifies the context encoder accordingly to enable it to encode this structure.

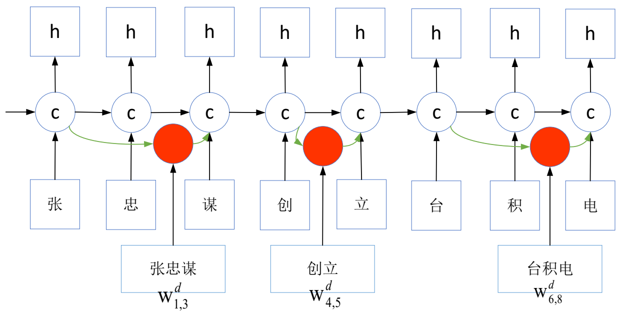


Figure 10. Lattice LSTM structure

Notes. 张忠谋 (Zhang Zhongmo) 创立 (founded) 台积电 (TSMC)

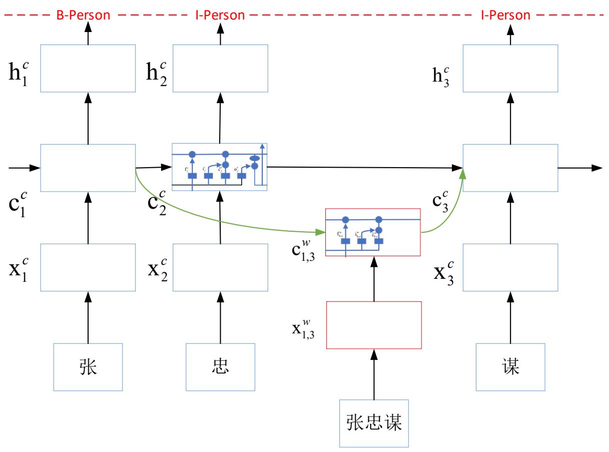


Figure 11. The structure diagram of Lattice LSTM based named entity recognition

As shown in Figure 10, the model automatically controls the information flow from the beginning to the end of a sentence, and dynamically routes the information from different paths to each character with a gating unit. After training on the corpus data, Lattice-LSTM can learn to find more useful words from the context to obtain better performance. Compared with LSTM-CRF based character sequence mark model and word sequence mark model, Lattice-LSTM improves the accuracy of entity recognition and has the best results in Chinese entity recognition data sets in different fields. Compared with character based and word based entity recognition methods, this model has the advantage of using explicit word information instead of character sequence marks, and does not have segmentation errors.

Figure 11 shows the structure of named entity recognition based on Lattice LSTM. The input of the model is a character sequence c_1, c_2, …, c_m, as well as all character subsequences matched from the dictionary, which is constructed by using large automatically segmented original text. Use $W^d_{b,e}$ to represent the subsequence starting with character index b and ending with character index e, as shown in Figure 10, $W^d_{1,3}$ refers to " 张忠谋 " and $W^d_{6,8}$ refers to " 台积电 ". The model involves four types of vectors, namely, input vector, output hidden vector, cell vector and gate vector (See Figure 11 & Figure 5). As a basic component, character input vectors x_j^c are used to represent each character c_j . The basic loop structure of the model is constructed using the character unit vector c_j^c and the hidden vector h_j^c of each c_j , which c_j^c is used to record the circular information flow from the beginning of the sentence to the c_j , and h_j^c is used to mark the CRF sequence. The basic loop LSTM formula is shown below.

$$\begin{bmatrix} i_j^c \\ o_j^c \\ f_j^c \\ \tilde{c}_j^c \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W^{c^T} \begin{bmatrix} x_j^c \\ h_{j-1}^c \end{bmatrix} + b^c \right)$$

$$c_j^c = f_j^c \odot c_{j-1}^c + i_j^c \odot \tilde{c}_j^c$$

$$h_j^c = o_j^c \odot \tanh\left(c_j^c\right)$$

where c_j^c , f_j^c and o_j^c represent a group of input gates, forgetting gates and output gates respectively. W^{c^T} and b^c is the model parameter. σ represents the sigmoid function. However, unlike the character based model, the calculation of c_j^c takes into account the dictionary subsequence $W^d_{b,e}$ in the sentence. Each subsequence is represented by a $x^w_{b,e} = e^w(W^d_{b,e})$ formula, where e^w represents the same word embedding. In addition, a word unit $c^w_{b,e}$ is used to represent the cyclic state $x^w_{b,e}$ starting from the sentence. The value of $c^w_{b,e}$ is calculated by the following formula.

$$\begin{bmatrix} i^w_{b,e} \\ f^w_{b,e} \\ \tilde{c}^w_{b,e} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W^{w^T} \begin{bmatrix} x^w_{b,e} \\ h^c_b \end{bmatrix} + b^w \right)$$

$$c^w_{b,e} = f^w_{b,e} \odot c^c_b + i^w_{b,e} \odot \tilde{c}^w_{b,e}$$

where $i^w_{b,e}$ and $f^w_{b,e}$ are a set of input gates and forgetting gates. Because tags are executed only at the character level, there is no output gate for the word cell. For $c^w_{b,e}$, there are more cyclic information paths into each c_j^c , for example, the input sources of c_7^c include x_8^c (electricity), $x^w_{6,8}$ (TSMC), link all $c^w_{b,e}$ and b to unit c_e^c , and use additional gate $i^w_{b,e}$ to control weight for each subsequence unit $c^w_{b,e}$.

$$i_{b,e}^{c} = \sigma\left(W^{l^T}\begin{bmatrix} x_e^c \\ c_{b,e}^w \end{bmatrix} + b^l\right)$$

Therefore, we have

$$c_j^c = \sum\nolimits_b \alpha_{b,j}^c \odot c_{b,j}^w + \alpha_j^c \odot \tilde{c}_j^w$$

Normalize the gate values $i_{b,i}^c$ and i_i^c to $\alpha_{b,i}^c$ and $\alpha_{c,i}^c$. The final hidden vector h_i^c is then calculated. During entity recognition training, the loss value is back propagated to the parameters W^c, b^c, W^w, b^w, W^l, and b^l, allowing the model to dynamically focus on tags with higher correlation. The standard CRF layer is applied to h_1, h_2, ..., h_τ, where τ is the *n* based on character and Lattice model or *m* based on word model. The probability of tag sequence $y = l_1, l_2, ..., l_\tau$ is

$$P(y \mid s) = \frac{\exp\left(\sum\nolimits_i \left(W_{CRF}^{li} h_i + b_{CRF}^{(l_{i-1},l_i)}\right)\right)}{\sum\nolimits_{y'} \exp\left(\sum\nolimits_i \left(W_{CRF}^{l'_i} h_i + b_{CRF}^{(l'_{i-1},l'_i)}\right)\right)}$$

where y' represents any tag sequence, W_{CRF}^{li} is a model parameter specific to l_i , and $b_{CRF}^{(li-1,\ li)}$ is a deviation specific to l_{i-1} and l_i . Finally, the first order Viterbi algorithm is used to find the tag sequence with the highest score in the word based or character based input sequence. Given a set of manually marked training data $\{(s_i, y_i)|_{i=1}^N\}$, the L2 regularized sentence level logarithm likelihood loss function is used to train the model:

$$L = \sum\nolimits_{i=1}^N \log\left(P(y_i \mid s_i)\right) + \frac{\lambda}{2}\|\Theta\|^2$$

where λ is the L2 regularization parameter, Θ representing the parameter set.

3.2.4 Model Training

The Lattice LSTM model originally uses four different datasets, including OntoNotes 4, MSRA, Weibo NER, and Chinese resume to train. We add our own datasets (Mine) to the catalog and train them separately. The statistical results of the datasets are shown in Table 2.

Table 2. Lattice LSTM training dataset

Datasets	Type	Training set	Validation set	Test set
OntoNotes	Sentence	15.7K	4.3k	4.3k
	Characters	491.9k	200.5k	208.1k
MSRA	Sentence	46.4k	--	4.4k
	Characters	2169.9k	--	172.6k
Weibo	Sentence	1.4k	0.27k	0.27k
	Characters	73.8k	14.5k	14.8k
Chinese resume	Sentence	3.8k	0.46k	0.48k
	Characters	124.1k	13.9k	15.1k
Mine	Sentence	7.2k	1.8k	1.8k
	Characters	134.9k	67.4k	67.2k

Regarding the model parameters, we used the Lattice-LSTM model fixed parameters, and Table 3 shows the hyper-parameter values of our model. The parameters were not modified except for the number of hidden layers changed to 100. The embedding size is set to 50 and the dropout is set to 0.5. The model is optimized using a random gradient descent method, with an initial learning rate of 0.015 and a decay rate of 0.05.

Table 3. Lattice LSTM experimental parameters

Parameter	Value	Parameter	Value
char emb size	50	bigram emb size	50
lattice emb size	50	LSTM hidden	100
char dropout	0.5	lattice dropout	0.5
LSTM layer	1	regularization λ	1e-8
learning rate lr	0.015	lr decay	0.05

After the initial establishment of the model, it is necessary to train the model and verify its performance. This article mainly evaluates the performance of the model using three indicators: Precision, Recall, and F1-Score. The F1 value belongs to the comprehensive performance indicator and is considered the most authoritative indicator. The detailed formula is as follows:

$$P = \frac{T}{N} \times 100\%$$

$$R = \frac{T}{M} \times 100\%$$

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\%$$

Among them, *T* represents the number of correct entity recognition, *N* represents the total number of entity recognition, and *M* represents the total number of labeled data.

3.2.5 Experimental Analysis

In our experiment, we divided it into two parts. In the first part, we first split our data (Mine) into a training set, a validation set, and a test set, and combined them with the model to conduct the experiment. Part 2: In order to verify the impact of specific domains on entity extraction, we will strengthen the dataset, and the enhanced dataset will be Mine-en, mainly adding domain entities. In the third part, we also compared our own dataset with the OntoNotes dataset and Weibo data. The experimental results are shown in Table 4.

Table 4. Experimental results of entity extraction

Datasets	P (%)	R (%)	F1 (%)
OntoNotes	75.87	70.35	72.87
Weibo	52.98	61.91	58.09
Mine	64.29	36.73	46.75
Mine-en	87.34	74.11	80.19

Experiment 1 shows that our dataset F1 can only achieve an effect of 46.75%, which is not as good as the OntoNotes dataset (72.87%) and Weibo dataset (58.09%) in terms of model performance. The reason for this is that specific domain datasets have fewer labels, resulting in imbalanced data. Experiment 2 shows that as the number of domain datasets increases, the performance of the model continues to improve. Based on the Lattice-LSTM model, we performed excellently on our semiconductor dataset, with F1 values increasing by 10% compared to the OntoNotes dataset and 38% compared to the Weibo dataset. The main reasons can be summarized as follows: (1) The specific domain dataset has a small physical range, strong professional terminology, and low noise. (2) There are few types of labels in the dataset, with obvious differentiation and low ambiguity. (3) The training dataset is large, encompassing the vast majority of entities in the field. We have shown the variation of F1 with the number of iterations in Figure 12.

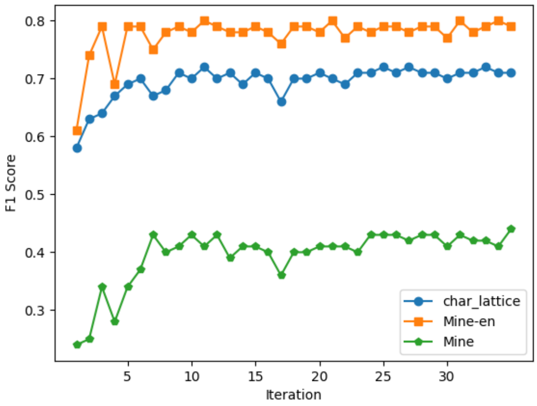


Figure 12. The relationship between F1 and iterative training

In summary, in the case of insufficient domain data, the extraction of domain entities is difficult. However, while ensuring data quality, training with a large amount of domain data can easily achieve a high level of model performance.

3.3 Relation Extraction

Through the above section, the entity recognition process has been completed, and the model has the ability to extract semiconductor entity information. This section mainly combines sentence information to extract corresponding relationships from entities. The combination of entity extraction and relationship extraction forms a complete knowledge extraction, which is transformed into structured data and saved text to build a knowledge graph of the semiconductor industry chain.

Most of the traditional relational extraction schemes use supervised learning. In the case of sufficient training data, this method can achieve good results, but supervised learning requires manual annotation to complete a large number of annotation tasks, which costs a lot. In order to solve this problem, our paper adopts the method of remote supervised learning, and a large number of labeled data are obtained from the existing information, but this method also introduces some noise.

Our paper is based on relationship extraction in the semiconductor field. The classification and type of relationships are clear, so the task of relationship extraction can be seen as relationship judgment based on entity extraction, whether there are relationships between entities and the categories of relationships. At the same time, in order to improve the impact of remote supervision data annotation, after coding using the Piecewise Convolutional Neural Networks (PCNN) [19] model, the attention mechanism at the package level is used to obtain information from sentences with high matching degree to reduce the impact of noise as much as possible.

3.3.1 Remote Supervision Label

In order to break the limitation of manual data annotation in supervised learning, Mintz et al. [20] proposed the Remote Supervision algorithm. The core idea of this algorithm is to align the text with the large-scale knowledge map, and label the text using the existing entity relationships in the knowledge map. The basic assumption of remote supervision is that if a triple (E1, R, E2) can be obtained from the knowledge map (E represents an entity, R represents a relationship) and E1 and E2 appear together in the sentence S, S expresses the relationship R between E1 and E2, which is labeled as a training positive example. Remote supervised algorithm is widely used in the mainstream relational extraction system. For example, Zhou et al. [21] proposed a remote supervised relationship extraction method with self-selective attention. This method uses a layer of convolution and self-attention mechanism to encode instances, to learn better semantic vector representation of instances.

The data in this chapter are annotated based on the data processed by entity recognition and the existing knowledge and remote supervision algorithm. First, filter the existing sentences, mainly filtering some invalid sentences that do not contain entities or only contain one entity, or the relationship between entities is not in the relationship extraction type. The remaining sentences are the original data of this experiment, which are annotated by remote supervision.

According to the above description of remote supervision, some noise will be introduced by using remote supervision labels, such as “Dr. Chen Zhikuan is the president and CEO of Xinsi Technology” (陈志宽博士任美国新思科技总裁兼联席首席执行官), which clearly indicates that Chen Zhikuan is the executive director of New Ideas Technology. However, the sentence “Chen Zhikuan led Xinsi Technology to donate the world’s top core chip design tools worth millions of dollars at that time to Tsinghua University” (陈志宽带领新思科技捐赠当时价值数百万美元的全球顶尖的核心芯片设计工具给清华大学). The specific relationship between Chen Zhikuan and Xinsi Technology was not clearly demonstrated. In order to reduce the influence of such noise, a piecewise convolutional neural network model based on sentence attention mechanism is used to extract relations.

3.3.2 Build Features

The input features of this experiment mainly include word features and location features. For word features, they have been introduced in this chapter. For Chinese relationship extraction, we first use the word segmentation tool Jieba to segment sentences to facilitate the subsequent determination of location features. After word segmentation, this experiment also uses Skip-gram model of Word2Vec to construct word features. For location features, after previous data processing, all sentences contain at least two entities, so it is important to judge the related words between the two entities. The relative position between sentences can be used as the key to judge the relationship between sentences. The closer words are to entities, the more likely they are to represent the relationship between entities. Therefore, this paper uses the relative position information of words as the feature of relation extraction [22]. For example, in the following example sentence, the relative positions from "CEO (首席执行官)" to "Chen Zhikuan (陈志宽)" and "Xinsi Technology (新思科技)" are 8 and 4 respectively. As shown in Figure 13.

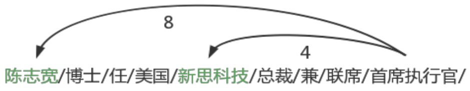


Figure 13. Relative position representation

3.3.3 Piecewise Convolution Neural Network Model

The network structure of the piecewise convolutional neural network (PCNN) model processes a sentence in four steps: feature representation, convolution, piecewise maximum pooling, and Softmax classification. In order to connect the attention mechanism, Softmax will be introduced later. As shown in Figure 14 below.

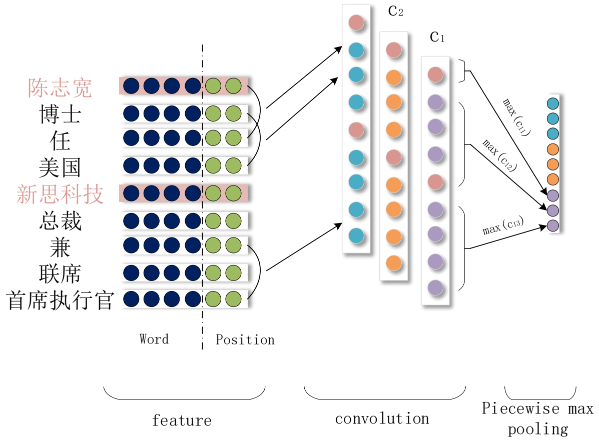


Figure 14. Piecewise convolution neural network model

(1) Feature representation

The input vector consists of two parts, aiming at the two types of features constructed previously: word features and location features. Word feature uses Word2Vec to generate word vector. The position feature transforms the relative distance into a real value vector by searching the position embedding matrix. Suppose that the size of word embedding is $dw = 4$, and the size of positional embedding is $dp = 1$. In compound word embedding and positional embedding, the vector representation part converts the instance into a matrix $S = R^{s \times d}$, where s is the sentence length, and $d = dw + dp * 2$. The matrix S is used as the input of the feature.

(2) Convolution

Assuming that the width of the convolution kernel is w (sliding window) and the length is d (feature vector dimension of words), the size of the convolution kernel is $W = w * d$. The step size is 1. Input layer is a $s \times d$ dimensional matrix, the convolution operation is to use the convolution kernel W and the w-gram of S to do the dot product every time it slides, and get the sequence $c \in R^{s+w-1}$.

$$c_j = wq_{j-w+1:j}$$

The value range of j is $1 < j < s + w - 1$, and the value outside the range is zero. In order to have the ability to capture different features, it is usually necessary to use multiple filters (or feature maps) in convolution. Suppose we use n filters ($W = \{w_1, w_2, \ldots, w_n\}$), the convolution operation can be expressed as

$$c_{ij} = w_i q_{j-w+1:j}, 1 \le i \le n$$

In this experiment $C = \{c_1, c_2, \ldots, c_n\} \in R^{n \times (s+w-1)}$, three different filters are used in the convolution process.

(3) Piecewise maximum pooling

Because a single maximum pooling will reduce the size of the hidden layer too quickly, and it is too rough to capture fine-grained features for relationship extraction. In addition, a single largest pool is not enough to capture the structure information between two entities. In order to capture the structure and other potential information, the experiment divides the convolution result into three parts according to the location of two given entities, and designs a piecewise maximum pooling layer. The piecewise maximum pooling process returns the maximum value in each segment, rather than a single maximum value in the entire sentence. Therefore, compared with traditional methods, it is expected to show better performance.

The vector c_i obtained by each convolution kernel is divided into three parts $\{c_{i1}, c_{i2}, c_{i3}\}$ according to two entities, and the maximum pooling by segments is to take the maximum value of each part respectively:

$$p_{ij} = \max(c_{ij}),\ 1 \le i \le n,\ 1 \le j \le 3$$

After subsection pooling, we can get a 3-dimensional vector $p_i = \{p_{i1}, p_{i2}, p_{i3}\}$. To facilitate the next input to the softmax layer, the pooled vector p_i of n convolution kernels is spliced into a $1 \times (3 * n)$ vector $p_{1:n}$, and finally, the tanh activation function is used for nonlinear processing to obtain the final output:

$$g = \tanh(p_{1:n})$$

The size of g is fixed and has nothing to do with the length of the sentence.

3.3.4 Constructing Package Level Attention Mechanism on Multiple Instances

To solve the problem of label errors, we construct package level attention on multiple instances, expecting to dynamically reduce the weight of those noisy instances.

Finally, the relation vectors weighted by package level attention are extracted.

An important challenge in relational classification is the interference of noise. The introduction of multi instance learning can alleviate the problem of error labels to some extent. After introducing multi instance learning, the objective function is still the cross entropy loss function, but the loss is calculated based on packets rather than sentences. We apply the softmax operation to all relationship types:

$$p(r\mid m_i^j;\theta)=\frac{e^{o_r}}{\sum_{k=1}^{n_r}e^{o_k}}$$

where m_i^j represents the input of the j^{th} sentence in the i^{th} package, θ is a network parameter and $\theta=(E, PF_1, PF_2, W, W_1)^2$, r represents the relationship type, o_r is the r^{th} component of PCNN model output. For each entity pair, it may not only appear in one sentence, but also multiple sentences (q_i). After full connection layer processing, multiple sentences containing the same entity pair can obtain different prediction tags and prediction probabilities. For the prediction results of each entity pair, the prediction tag of the sentence with the highest prediction probability is selected as the prediction tag of the entity pair, and used to calculate the cross entropy loss. The specific formula is as follows:

$$j^*=\arg_i\max p\left(y_i\mid m_i^j;\theta\right),1\le j\le q_i.$$

If there are T entity pairs, it is necessary to select T sentences and calculate the cross entropy loss:

$$J(\theta)=\sum_{i=1}^{T}\log p(y_i\mid m_i^j;\theta)$$

Finally, the gradient is obtained by gradient descent method, and the error is back propagated.

Through the previous introduction, we know that the traditional back propagation algorithm modifies the network according to all training cases, while the multi instance learning back propagation algorithm modifies the network according to packets. Therefore, our method captures the essence of remote supervised relationship extraction, and some training examples will inevitably be mislabeled. When a trained PCNN is used for prediction, a packet is marked as positive when and only when the output of the network is assigned as a positive mark on at least one instance of the network.

In the training and prediction phase, the method based on multi instance learning only uses the sentence information with the maximum probability for each entity pair, which will inevitably ignore the valuable sentence semantic information. The introduction of package level attention mechanism can solve this problem well [23]. By constructing the sentence selection attention mechanism, the valuable sentences are given a higher weight, while the noisy sentences are given a smaller weight, and then the sum is accumulated according to the weight as the feature vector of the whole package. The structure of the package with attention mechanism is shown in Figure 15. Set s is the weighted sum of these sentence vectors.

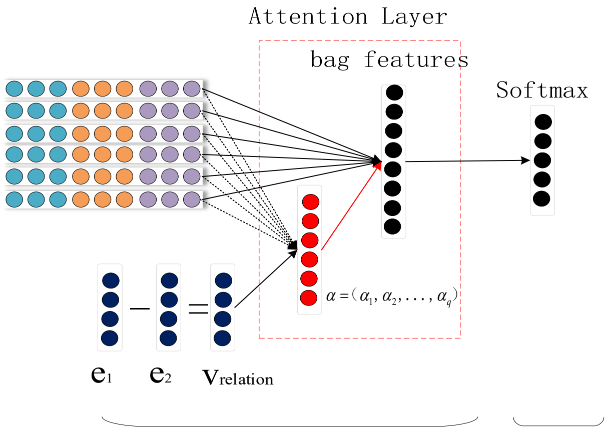


Figure 15. Package with attention mechanism

$$s=\sum_i \alpha_i x_i$$

where α_i is the weight of each sentence vector x_i. Sentences that use selective attention to reduce noise. Therefore, α_i is further defined as:

$$\alpha_i=\frac{\exp(e_i)}{\sum_k \exp(e_k)}$$

where e_i is a query based function, which scores the matching degree of input statement x_i and prediction relation r. We choose the bilinear form, which can achieve the best performance in different schemes.

$$e_i=x_iAr$$

where A is a weighted diagonal matrix and r is a query vector related to relation r, representing the representation of relation r. Finally, we define the conditional probability through a softmax layer as follows.

$$p\left(r\mid S,\theta\right)=\frac{e^{o_r}}{\sum_{k=1}^{n_r}e^{o_k}}$$

where n_r is the total number of relationships, and o is the final output of the neural network, corresponding to the scores related to all relationship types, as defined below.

$$o=Ms+d$$

where $d\in R^{n_r}$ is the offset vector and M is the representation matrix of the relationship.

3.3.5 Piecewise Convolution Neural Network Based on Package Level Attention Mechanism

First, PCNN is used to extract the eigenvector v of each sentence, and then the attention weight of each sentence is calculated by stitching the hidden layers. Finally, the weighted sum of all sentence feature vectors is the feature of the package. The overall framework is shown in Figure 16 below:

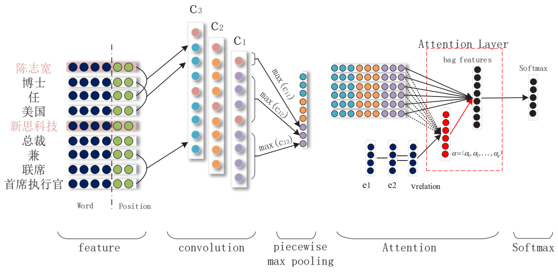


Figure 16. Piecewise convolution neural network with sentence level attention mechanism

3.3.6 Experimental Analysis

This article will use the PCNN model from the OpenNRE toolkit for relationship extraction. OpenNRE is an open-source neural network relationship extraction toolkit that allows us to directly call APIs for relationship extraction, greatly simplifying our work. By adjusting, we used Table 5 as our experimental parameters.

Table 5. PCNN experimental parameters

Parameter	Value	Parameter	Value
Hidden size	230	Word size	50
Position size	50	Kernel size	3
Padding size	1	dropout	0

We filter out sentences that satisfy two entities in the Mine-en dataset to form a new relational dataset Mine-re, and train the relational dataset Mine-re in the model. We compare it with the NYT10m dataset to check its effectiveness. The results are shown in Table 6.

Table 6. Experimental results of relationship extraction

Data set	F1
NYT10m	62.45
Mine-re	66.67

The experiment shows that our dataset performs well on the PCNN model, which is more stable than the NYT10m dataset, and the performance is improved to 66.67%. The reason for this is that the domain dataset has clear entities and clear relationships. The data collected this time is all from the semiconductor field, and pipeline based relationship extraction is used. The previous entity extraction has determined that the entity belongs to the field, and the model did not receive much interference during relationship extraction. However, our F1 value is smaller than expected, which may be due to our small dataset.

4 Knowledge Graph Construction

In this section, Neo4j graph database will be used to store those entities and relationship for semiconductor industry chain. Logically, we usually divide the knowledge graph into two layers: data layer and pattern layer. Pattern layer is above the data layer. It is the core of the knowledge graph and stores refined knowledge and is usually managed through the ontology library. Regarding the data layer, it stores entities and real data.

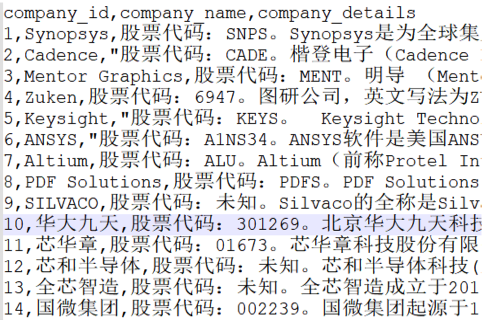
4.1 Analysis and Construction of Semiconductor Domain Knowledge Database

4.1.1 Domain Knowledge Database Analysis

The whole process of chip production divides into 15 parts including EDA software, IP modules, masks, lithography materials, deposition materials, silicon materials, etching materials, polishing materials, special gases, epitaxial wafers, packaging materials, semiconductor equipment, IC design, IC manufacturing, IC packaging and testing. The extracted data are classified to ensure the hierarchical correspondence. After establishing the correct category and hierarchy system, hierarchical division is carried out for the attribute of each category ontology and the relationship between ontology.

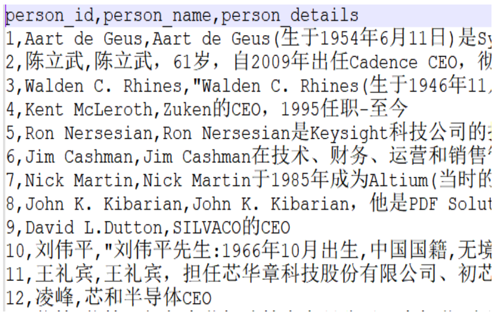
4.1.2 Ontology Division and Construction

We divide the types of entities into five types, corresponding to company name (Figure 17(a)), CEO name (Figure 17(b)), product name (Figure 17(c)), industrial chain port and application domain (Figure 17(d)). The relationship types are divided into four types: person and company (Figure 17(e)), company and product (Figure 17(f)), product and industry chain (Figure 17(g)), and industry to domain (Figure 17(h)). The connection between each entity will be created according to the relationship extracted from the data.



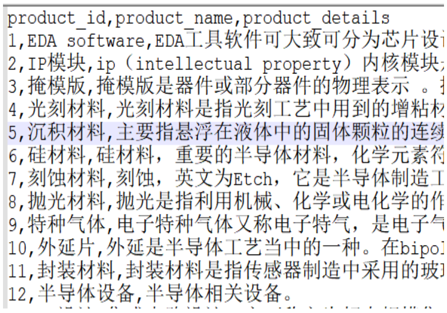
```
company_id,company_name,company_details
1,Synopsys,股票代码: SNPS。Synopsys是为全球集
2,Cadence,"股票代码: CADE。楷登电子（Cadence
3,Mentor Graphics,股票代码: MENT。明导 （Ment
4,Zuken,股票代码: 6947。图研公司, 英文写法为Z
5,Keysight,"股票代码: KEYS。  Keysight Techno
6,ANSYS,"股票代码: A1NS34。ANSYS软件是美国ANS
7,Altium,股票代码: ALU。Altium（前称Protel In
8,PDF Solutions,股票代码: PDFS。PDF Solutions
9,SILVACO,股票代码: 未知。Silvaco的全称是Silv
10,华大九天,股票代码: 301269。北京华大九天科技
11,芯华章,股票代码: 01673。芯华章科技股份有限
12,芯和半导体,股票代码: 未知。芯和半导体科技(
13,全芯智造,股票代码: 未知。全芯智造成立于201
14,国微集团,股票代码: 002239。国微集团起源于1
```

(a) company.csv



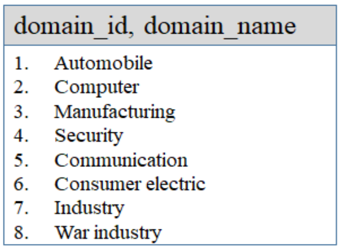
```
person_id,person_name,person_details
1,Aart de Geus,Aart de Geus(生于1954年6月11日)是Sy
2,陈立武,陈立武, 61岁, 自2009年出任Cadence CEO, 彻
3,Walden C. Rhines,"Walden C. Rhines(生于1946年11,
4,Kent McLeroth,Zuken的CEO, 1995任职-至今
5,Ron Nersesian,Ron Nersesian是Keysight科技公司的
6,Jim Cashman,Jim Cashman在技术、财务、运营和销售
7,Nick Martin,Nick Martin于1985年成为Altium(当时的
8,John K. Kibarian,John K. Kibarian, 他是PDF Solut
9,David L.Dutton,SILVACO的CEO
10,刘伟平,"刘伟平先生:1966年10月出生,中国国籍,无境
11,王礼宾,王礼宾, 担任芯华章科技股份有限公司、初芯
12,凌峰,芯和半导体CEO
```

(b) CEO.csv



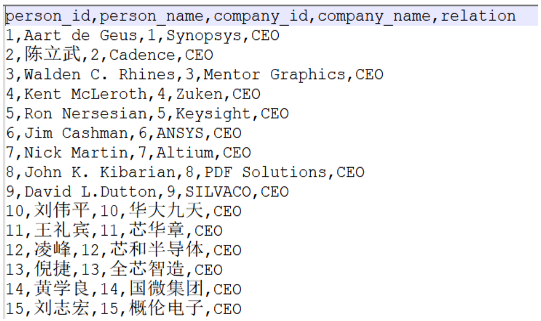
```
product_id,product_name,product_details
1,EDA software,EDA工具软件可大致可分为芯片设
2,IP模块,ip（intellectual property）内核模块
3,掩模版,掩模版是器件或部分器件的物理表示 。
4,光刻材料,光刻材料是指光刻工艺中用到的增粘
5,沉积材料,主要指悬浮在液体中的固体颗粒的连
6,硅材料,硅材料，重要的半导体材料，化学元素
7,刻蚀材料,刻蚀，英文为Etch，它是半导体制造
8,抛光材料,抛光是指利用机械、化学或电化学的
9,特种气体,电子特种气体又称电子特气，是电子
10,外延片,外延是半导体工艺当中的一种。在bipo
11,封装材料,封装材料是指传感器制造中采用的玻
12,半导体设备,半导体相关设备。
```

(c) product.csv



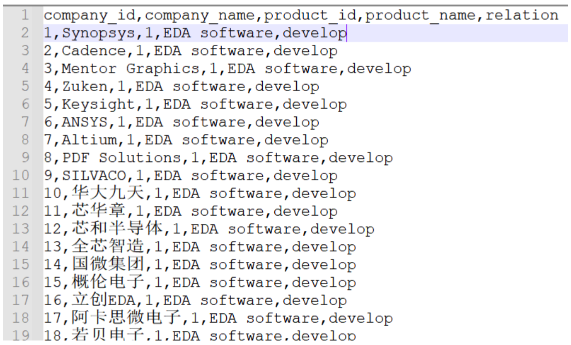
```
domain_id, domain_name
1.    Automobile
2.    Computer
3.    Manufacturing
4.    Security
5.    Communication
6.    Consumer electric
7.    Industry
8.    War industry
```

(d) domain.csv



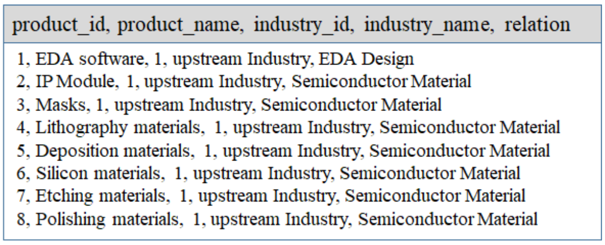
```
person_id,person_name,company_id,company_name,relation
1,Aart de Geus,1,Synopsys,CEO
2,陈立武,2,Cadence,CEO
3,Walden C. Rhines,3,Mentor Graphics,CEO
4,Kent McLeroth,4,Zuken,CEO
5,Ron Nersesian,5,Keysight,CEO
6,Jim Cashman,6,ANSYS,CEO
7,Nick Martin,7,Altium,CEO
8,John K. Kibarian,8,PDF Solutions,CEO
9,David L.Dutton,9,SILVACO,CEO
10,刘伟平,10,华大九天,CEO
11,王礼宾,11,芯华章,CEO
12,凌峰,12,芯和半导体,CEO
13,倪捷,13,全芯智造,CEO
14,黄学良,14,国微集团,CEO
15,刘志宏,15,概伦电子,CEO
```

(e) personToCompany.csv



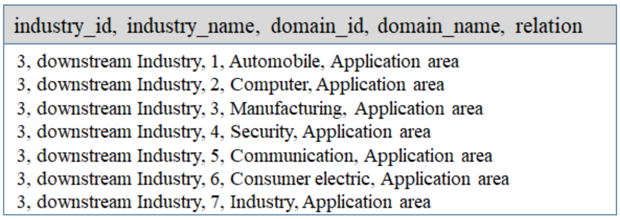
```
company_id,company_name,product_id,product_name,relation
1,Synopsys,1,EDA software,develop
2,Cadence,1,EDA software,develop
3,Mentor Graphics,1,EDA software,develop
4,Zuken,1,EDA software,develop
5,Keysight,1,EDA software,develop
6,ANSYS,1,EDA software,develop
7,Altium,1,EDA software,develop
8,PDF Solutions,1,EDA software,develop
9,SILVACO,1,EDA software,develop
10,华大九天,1,EDA software,develop
11,芯华章,1,EDA software,develop
12,芯和半导体,1,EDA software,develop
13,全芯智造,1,EDA software,develop
14,国微集团,1,EDA software,develop
15,概伦电子,1,EDA software,develop
16,立创EDA,1,EDA software,develop
17,阿卡思微电子,1,EDA software,develop
18,若贝电子,1,EDA software,develop
```

(f) companyToProduct.csv



```
product_id, product_name, industry_id, industry_name, relation
1, EDA software, 1, upstream Industry, EDA Design
2, IP Module, 1, upstream Industry, Semiconductor Material
3, Masks, 1, upstream Industry, Semiconductor Material
4, Lithography materials, 1, upstream Industry, Semiconductor Material
5, Deposition materials, 1, upstream Industry, Semiconductor Material
6, Silicon materials, 1, upstream Industry, Semiconductor Material
7, Etching materials, 1, upstream Industry, Semiconductor Material
8, Polishing materials, 1, upstream Industry, Semiconductor Material
```

(g) productToIndustry.csv



```
industry_id, industry_name, domain_id, domain_name, relation
3, downstream Industry, 1, Automobile, Application area
3, downstream Industry, 2, Computer, Application area
3, downstream Industry, 3, Manufacturing, Application area
3, downstream Industry, 4, Security, Application area
3, downstream Industry, 5, Communication, Application area
3, downstream Industry, 6, Consumer electric, Application area
3, downstream Industry, 7, Industry, Application area
```

(h) IndustryToDomain.csv

Figure 17. Entity and relationship data

4.2 Knowledge Graph Storage of Semiconductor Industry Chain Based on Neo4j

4.2.1 Method for Creating Neo4j Knowledge Map

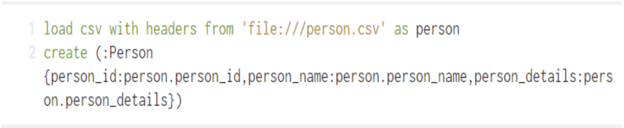
At present, Neo4j has three versions. For research use, we use the community version here. After downloading, execute the DOS command "neo4j.bat console" in the bin directory. The neo4j.bat console command opens the Neo4j interface as required and creates a semiconductor project.

4.2.2 Build a KG

Neo4j uses the graph query language Cypher to build the knowledge graph according to the above graph analysis. The construction process can be divided into three processes.

(1) Entity Creation

The creation of entity classes mainly includes <Person> class, <Company> class, <Product> class, <Industry> class and <Domain> class. Each class consists of several entities, which can be constructed in batches according to the csv files extracted above. Some codes are shown in Figure 18.

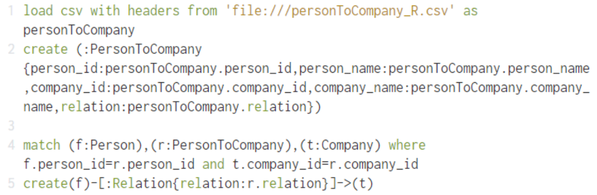


```
load csv with headers from 'file:///person.csv' as person
create (:Person
{person_id:person.person_id,person_name:person.person_name,person_details:pers
on.person_details})
```

Figure 18. Entity creation partial code

(2) Relationship Creation

The creation of relationship classes mainly includes <personToCompany> class, <companyToProduct> class, <productToIndustry> class, and <industryToDomain> class. The way to create a relationship is slightly different from that of an entity, mainly because there is an additional connection step. Some codes are shown in Figure 19.



```
load csv with headers from 'file:///personToCompany_R.csv' as
personToCompany
create (:PersonToCompany
{person_id:personToCompany.person_id,person_name:personToCompany.person_name
,company_id:personToCompany.company_id,company_name:personToCompany.company_
name,relation:personToCompany.relation})

match (f:Person),(r:PersonToCompany),(t:Company) where
f.person_id=r.person_id and t.company_id=r.company_id
create(f)-[:Relation{relation:r.relation}]->(t)
```

Figure 19. Relationship creation partial code

(3) Connection Creation

The establishment of the connection is mainly to form a completed system. The semiconductor industrial chain system is mainly divided into upstream, midstream and downstream. An IndustryChain class is created here as required to connect the entire industry. The specific method is the same as above. The partial screenshot of knowledge map of semiconductor industry chain is shown in Figure 20.

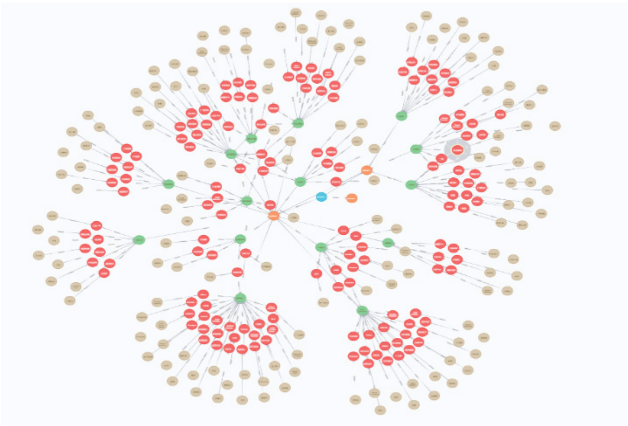


Figure 20. Partial screenshot of knowledge map of semiconductor industry chain

5 KG Query System Construction

In this section, we will construct a KG prototype query system of semiconductor industry chain. This system is developed by adopting the technical framework of springroot+Neo4j+D3.js.

5.1 System Architecture

Figure 21 is the System architecture. Since this is a prototype system, the Overall architecture development is on the local computer only.

Front end UI: This layer is mainly for the design of various interfaces, providing users with visual interactive operation interfaces, mainly for the design of login interfaces and query interfaces.

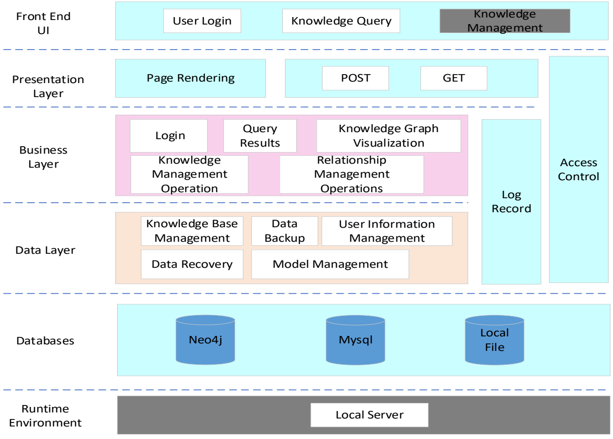


Figure 21. System architecture

Display layer: Data display, providing users with different interfaces, mainly for page rendering and access to front and back interfaces.

Business layer: The business layer is located between the data layer and the presentation layer, which serves as a link between the preceding and the following for data exchange.

Data layer: Its function is mainly responsible for database access, which can access database systems, binary files, and text documents to implement the operations of adding, deleting, modifying, and querying data tables.

Database: This layer mainly consists of MySql database, Neo4j graph database and local files. MySql is responsible for storing user information or some common structured data. Neo4j is mainly responsible for the storage of knowledge graph entities, relationships and corresponding attributes. The local file mainly stores some related text or picture information.

5.2 System Functions

The function of this system mainly consists of three parts: user login, knowledge query and knowledge management. The first part is user authentication. Users who pass the authentication can enter the knowledge graph page. The second part is the knowledge acquisition of different semiconductor modules, and the corresponding information is obtained by selecting different modules.

5.2.1 Login Function

The login function is shown in the Figure 22. This function uses the form verification and buttons provided by the "element ui" to design the login interface and introduces the "vue-i18n" component to realize the language switching function. Ordinary users can only use the query module, and administrator users can add, delete, and modify special permissions.

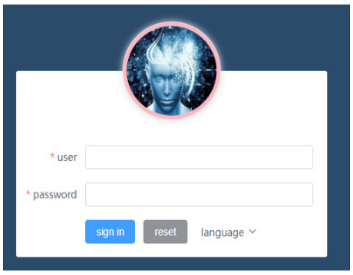


Figure 22. Login interface

5.2.2 Inquiry Function

The query function of the system is the main function of the system. This function is mainly realized by using the pull-down menu provided by the <element UI>, which provides a control method. Users can select requirements by clicking buttons. This function comprehensively analyzes the needs of users in different industries for system query from 15 aspects. Through the list classification of module selection in Figure 23, users can query the required module information according to their needs.

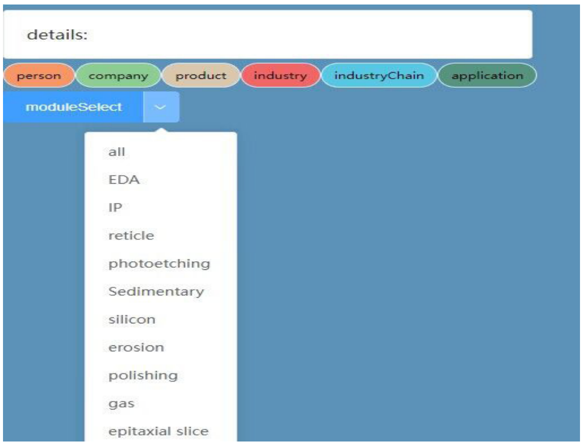
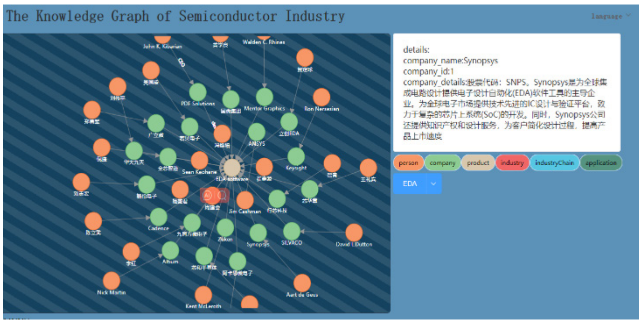

details:

Figure 23. List classification of module selection

Figure 24 shows the EDA software module. It shows all the EDA production companies and their representatives. The figure is mainly presented via <D3.js>. It also realizes the scaling function of the interface and the stretching of nodes, which is convenient for users to operate on the graph. There are corresponding relationships between nodes, such as CEO and development.

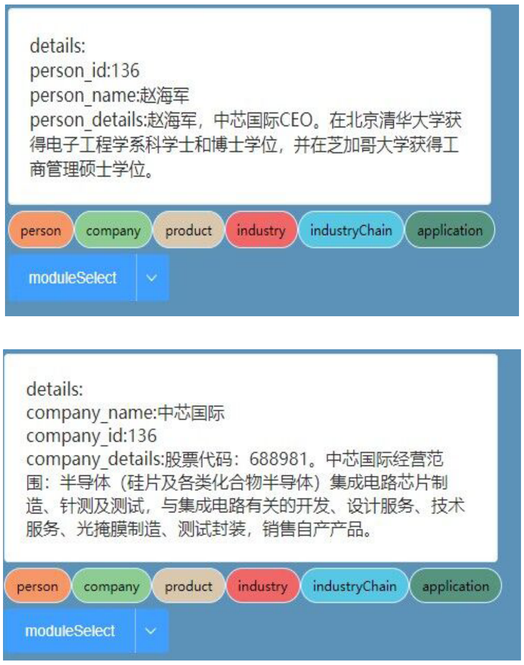

The Knowledge Graph of Semiconductor Industry

details:
company_name:Synopsys
company_id:1
company_details:股票代码：SNPS。Synopsys是为全球集成电路设计提供电子设计自动化(EDA)软件工具的主导企业。为全球电子市场提供技术先进的IC设计与验证平台，致力于复杂的芯片上系统(SoC)的开发。同时，Synopsys公司还提供知识产权和设计服务，为客户简化设计过程，提高产品上市速度

Figure 24. EDA software module

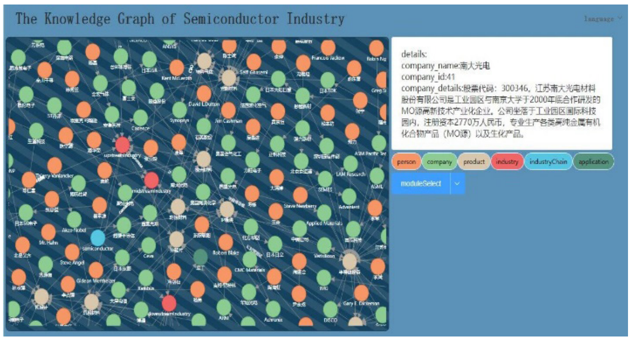
For different nodes, we use different colors to distinguish and prompt in the display bar, so that users can more clearly distinguish the meaning of each node. At the same time, in order to enable users to obtain more knowledge information, text display of specific information labeling is set for person nodes and company nodes. The user can click the node to view the specific information in the right display bar, which is mainly displayed in three parts, namely, node ID, name, and details. Users can not only obtain the names of people and companies, but also understand the profiles of people and the stocks and business scope of the company, so that users can understand the information of each node in more detail. The query interface is shown in Figure 25.

In addition, we have visualized the entire semiconductor industry chain, as shown in Figure 26 that is a display of Chinese and English interfaces. The semiconductor industry process in the figure is composed of upstream, midstream and downstream, and the industrial process is composed of red nodes. At the same time, the modules contained in the upstream are connected by upstream nodes, such as semiconductor equipment and materials; The midstream includes the following modules: IC design, packaging and testing; Downstream is represented directly by the application field. Each product may be produced by multiple companies. The company node contains the company name and details. You can click the corresponding company node to query, or view the information of the corresponding person and downstream information.


details:
person_id:136
person_name:赵海军
person_details:赵海军，中芯国际CEO。在北京清华大学获得电子工程学系科学士和博士学位，并在芝加哥大学获得工商管理硕士学位。

details:
company_name:中芯国际
company_id:136
company_details:股票代码：688981。中芯国际经营范围：半导体（硅片及各类化合物半导体）集成电路芯片制造、针测及测试，与集成电路有关的开发、设计服务、技术服务、光掩膜制造、测试封装，销售自产产品。

Figure 25. Inquiry UI

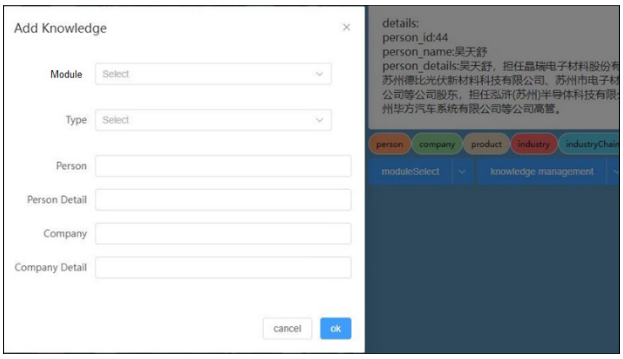

The Knowledge Graph of Semiconductor Industry

details:
company_name:南大光电
company_id:41
company_details:股票代码：300346。江苏南大光电材料股份有限公司是工业园区与南京大学于2000年底合作研发的MO源高新技术产业化企业。公司坐落于工业园区国际科技园内，注册资本2770万人民币，专业生产各类高纯金属有机化合物产品（MO源）以及生化产品。

Figure 26. Knowledge graph of semiconductor industry chain

5.2.3 Knowledge Management Function

The knowledge management module of the system is the special authority of the administrator, which mainly prevents the system errors caused by other users' lack of domain knowledge and misoperation, and also facilitates the administrator to operate the system more conveniently. This module pops up a dialog box in the style of a form. In order to stock in the newly created node, you need to fill in not only the node information, but also the module information. This system only provides a simple design to facilitate understanding the form of knowledge management operation on the web page. Similarly, you can add other functions or statistical reports you need to enrich the system. The knowledge adding interface is shown in Figure 27.



Add Knowledge

Module: Select

Type: Select

Person:

Person Detail:

Company:

Company Detail:

details:
person_id:44
person_name:吴天舒
person_details:吴天舒，担任晶瑞电子材料股份有
苏州德比光伏新材料科技有限公司、苏州市电子材
公司等公司股东，担任泓浒(苏州)半导体科技有限
州华方汽车系统有限公司等公司高管。

Figure 27. Knowledge adding UI

6 Conclusion and Future Works

Knowledge graph can provide a better form of knowledge management. Especially in a specific field, the knowledge graph is of great assistance to professionals. This paper using KG technology to develops a KG system for semiconductor industry chain. The main research work is summarized as follows.

(1) This paper has completed the collection and annotation of data and created a knowledge database in the semiconductor field. First, we completed the construction of the original data set from the data of major semiconductor websites and laboratories through data acquisition technologies. Then the original data set is clarified and the entity dictionary library is constructed manually. The word segmentation and part of speech tagging are performed using the Jieba word segmentation tool, and the data tagging is performed using HanLP. At the same time, the manual verification is performed and completed the construction of input features.

(2) This paper studies the implementation of entity extraction in semiconductor field based on Lattice-LSTM model. Due to the difference of language characteristics between Chinese and English, Chinese named entity recognition is relatively difficult. Lattice-LSTM is designed to solve this problem. Lattice-LSTM forms a new vector representation for each word node and sends the word vector to the context encoder, and modifies the context encoder accordingly to enable it to encode this structure. Lattice-LSTM model has been proved to be effective for Chinese entity recognition, and it performs well on various Chinese data sets.

(3) In this paper, we study the relationship extraction between semiconductor entities based on the piecewise convolutional neural network model of sentence attention mechanism. First, we use the method of remote supervision of data annotation to improve the efficiency of data annotation, but remote supervision will also bring about the impact of noise, so this paper uses piecewise convolutional neural network to improve the model. At the same time, in order to obtain the relationship between entities as much as possible, we use sentence attention mechanism to obtain information.

(4) This paper studies and develops a knowledge mapping system for semiconductor industry chain. We save the extracted relationships and entities in local files, conduct auxiliary review through manual verification, and then save the data in the graph database Neo4j. Based on the SpringBoot+Vue+Neo4j technical framework, this paper completes the development of the semiconductor industry chain knowledge graph system, and realizes the functions of knowledge query, visual display, addition, modification, deletion, etc. The use of this system can provide users with more convenient and intuitive services, and high practical value.

In this paper, a complete knowledge graph of the semiconductor industry chain has been constructed, so that users can easily obtain knowledge of the semiconductor industry. Although the accuracy of the model selected in this paper is very high, manual verification has to be assisted on the acquired data. In addition, in the process of entity extraction and relationship extraction, manual proofreading is required to ensure that the data in each link is as accurate as possible. In the future work, we will consider the use of domain specific models to build domain knowledge. At the same time, it will also consider improving domain knowledge and systems, such as chip design knowledge, chip manufacturing technology knowledge, and chip packaging and testing technology knowledge etc., to provide users with more services. It can also combine the construction of the knowledge graph with big data technology to establish a more complete semiconductor knowledge graph.

Acknowledgments

The paper was partially sponsored under grant NSTC 112-2221-E-141-005 by the National Science and Technology Council, Taiwan, R.O.C.

References

[1] A. Maedche, S. Staab, Ontology Learning for the Semantic Web, *IEEE Intelligent Systems*, Vol. 16, No. 2, pp. 72-79, March-April, 2001.

[2] S. L. Yang, R. Z. Han, Analysis of Foreign Methods and Tools of Mapping Knowledge Domain, *Document, Information & Knowledge*, No. 6, pp. 101-109, November, 2012.

[3] K. Börner, C. Chen, K. W. Boyack, Visualizing knowledge domains, *Annual Review of Information Science and Technology*, Vol. 37, No. 1, pp. 179-255, 2003.

[4] S. R. Eddy, Hidden Markov models, *Current Opinion in Structural Biology*, Vol. 6, No. 3, pp. 361-365, June, 1996.

[5] Book Review, Maximum-Entropy Models in Science and Engineering. by J. N. Kapur, *Biometrics*, Vol. 48, No. 1, pp. 333-334, March, 1992.

[6] A. McCallum, D. Freitag, F. C. Pereira, Maximum entropy Markov models for information extraction and segmentation, *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, Standord, CA, USA, 2000, pp. 591-598, June, 2000.

[7] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B.

Scholkopf, Support vector machines, *IEEE Intelligent Systems and their applications*, Vol. 13, No. 4, pp. 18-28, July-August, 1998.

[8] C. Sutton, A. McCallum, An introduction to conditional random fields, *Foundations and Trends® in Machine Learning*, Vol. 4, No. 4, pp. 267-373, April, 2012.

[9] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, *Deep captioning with multimodal recurrent neural networks (m-rnn)*, arXiv preprint arXiv:1412.6632, December, 2014. https://arxiv.org/abs/1412.6632

[10] J. Cross, L. Huang, *Incremental parsing with minimal features using bi-directional LSTM*, arXiv preprint arXiv:1606.06406, June, 2016. https://arxiv.org/abs/1606.06406

[11] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN, *Expert Systems with Applications*, Vol. 72, pp. 221-230, April, 2017.

[12] C. Sun, Z. Yang, L. Wang, Y. Zhang, H. Lin, J. Wang, Biomedical named entity recognition using BERT in the machine reading comprehension framework, *Journal of Biomedical Informatics*, Vol. 118, Article No. 103799, June, 2021.

[13] A. Goyal, V. Gupta, M. Kumar, A deep learning-based bilingual Hindi and Punjabi named entity recognition system using enhanced word embeddings, *Knowledge-Based Systems*, Vol. 234, Article No. 107601, December, 2021.

[14] Y. Zhang, P. Qi, C. D. Manning, Graph convolution over pruned dependency trees improves relation extraction, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2205-2215.

[15] Y. Zhao, H. Wan, J. Gao, Y. Lin, Improving relation classification by entity pair graph, *Proceedings of The Eleventh Asian Conference on Machine Learning*, Vol. 101, pp. 1156-1171, 2019.

[16] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735-1780, November, 1997.

[17] A. Graves, N. Jaitly, A. R. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, *2013 IEEE workshop on automatic speech recognition and understanding*, Olomouc, Czech Republic, 2013, pp. 273-278.

[18] Y. Zhang, J. Yang, *Chinese NER using lattice LSTM*, July, 2018. arXiv preprint arXiv:1805.02023. https://arxiv.org/abs/1805.02023

[19] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, *Proceedings of the 2015 conference on empirical methods in natural language processing*, Lisbon, Portugal, 2015, pp. 1753-1762.

[20] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, 2009, pp. 1003-1011.

[21] Y. Zhou, L. Pan, C. Bai, S. Luo, Z. Wu, Self-selective attention using correlation between instances for distant supervision relation extraction, *Neural Networks*, Vol. 142, pp. 213-220, October, 2021.

[22] Y. P. Zhang, *Research on the Construction Technology of Military Equipment Knowledge Graph*, Master's Thesis, Xidian University, Xi'an, China, 2021.

[23] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers*, Berlin, Germany, 2016, pp. 2124-2133.

Biographies

Charles Chen is current a Professor in School of Business, Minnan Normal University, China. His research interests include knowledge graphics, pattern recognition, machine learning, data mining, information system innovation, intelligent manufacturing, image process, intelligent wastewater treatment process control system and quantitative analysis.

Sai-Sai Shi is current a M.S. candidate student in School of Computer Science and Engineering, Minnan Normal University, China. His main research interests include knowledge graph and named entity recognition.

Sheng-Lung Peng is a Professor at the Department of Creative Technologies and Product Design, and the Dean of the College of Innovative Design and Management, National Taipei University of Business in Taiwan. He is also an honorary Professor at Beijing Information Science and Technology University, a visiting Professor at Ningxia Institute of Science and Technology in China, an adjunct Professor at National Dong Hwa University in Taiwan and Kazi Nazrul University in India. His research interests are algorithm design in the fields of artificial intelligence, bioinformatics, combinatorics, data mining, and networking.