Introduction of Intelligent Broker between User and Search Engine for Patent Retrieval from Internet

Dr. Dawei Pan
Mr. Dasheng Chu
Issues Related to Searching Internet

- keyword search
  - context, synonym, different format of a word
- analysis leading to grouping and categorization
- programming repeating search path
- customizing presentation of search result
Introduction of Intelligent Broker Architecture

Class Library → Intelligent Brokers → User Interface

Model-View-Controller Model
Class Library
Intelligent Broker

<<class>>
IntelligentBroker

transient runner: Thread;
transient listeners: Vector;
private change:
PropertyChangeSupport;
……

- addIntelligentBrokerListener();
- removeIntelligentBrokerListener();
- notifyIntelligentBrokerListener();
- IntelligentBrokerEventFired();
- run();
- go();
- stop()
……

<<interface>>
Runnable

<<interface>>
Serializable

<<interface>>
IntelligentBrokerEventListener

+ IntelligentBrokerEventFired()

<<interface>>
IntelligentBrokerEvent

EventObject
Characteristics of Intelligent Broker

- JavaBeans Compliance
  - name convention
  - serializable
  - delegation event model
  - PropertyChangeSupport

- Thread and synchronized
User Interface
Application

• Patent Search

IntelligentBroker

FrontDesk
- addBroker(Broker)

InternetAccess
- addBroker(Broker)
- InternetAccess(URL)

Others

......
AI Strategies

- Reference Search
- Keyword Search
- Analysis
Reference Relationship

A Tree of Patent References of patent 5677435

It requires a special search algorithm.
Reference Search

Algorithm

\[
P_{\text{diff}} = \{ p_{\text{root}} \}, \quad P_{\text{ref}, 0} = \{ p_{\text{root}} \}, \quad P_{\text{ref}} = \emptyset, \quad n = 0, \quad i = 0
\]

\[
P_{\text{temp}} = \{ p | p \in P_{\text{diff}} \}, \quad P_{\text{temp}} = \{ p | p = \text{reference patents of } P_{\text{temp}} \}
\]

\[
P_{\text{ref}, i+1} = P_{\text{ref}, i} \cup P_{\text{temp}}
\]

\[
P_{\text{visited}} = P_{\text{visited}} \cup \{ p_{\text{temp}} \}
\]

\[
P_{\text{diff}} = P_{\text{ref}, i} - P_{\text{visited}}
\]

\[
P_{\text{diff}} = \emptyset?
\]

\[
P_{\text{diff}} = P_{\text{ref}, i+1} - P_{\text{visited}}
\]

\[
P_{\text{diff}} = \emptyset?
\]

\[
P_{\text{ref}} = \cup P_{\text{ref}, i}
\]
Constrained Reference Search

- Forward and Backward search control
- Depth control
- Field value constraints
- Counting quotation number
Criteria for Qualified Keyword

- not too highly frequent occurrence (thus article, conjunctive, preposition etc., must be eliminated.)
- high frequency
- within domain knowledge
Features of New Keyword Search

- Keyword originates from the root patent/article.
- Filtering and conversion.
- Build a set of qualified keywords.
- Count the frequency of keyword.
- Query is constructed with multiple qualified keywords and operator “and”.
- Multiple queries.
Keyword Search Algorithm

- Retrieve the target patent

  \[ P_{kw, temp} = \{ p | p \text{ is word included in the patent} \} \]

- Keyword conversion function

  \[ f: P_{kw, temp} \rightarrow P_{kw} \]

- \[ P_{kw, temp} = P_{kw} \]
  \[ P_{kw} = P_{kw, temp} \setminus P_{noise} \]

- \[ f_{freq} = \{ <x, y> | x \in P_{kw} \land y \in P_{freq} \} \]
  (a set of ordered pairs of keyword and its frequency in the patent)

- Query = “\( p_1 \land \cdots \land p_i \land \cdots \land p_n \)” (\( p_i \in \text{dom } f_{freq} \), \( n \) is the number of keywords, \( p_1 \) to \( p_n \) have corresponding higher frequency)

- \[ P_{pn, temp} \neq \emptyset \]?

  Yes

  \[ P_{pn} = P_{pn, temp} \cup P_{pn} \]

- No

- \[ P_{pn, temp} = \emptyset \]?

  Yes

  Other set

- No
Analysis

- Calculate the matching-rate between the keyword sets from any two patents.
- Find the context of the seed word.
- Matching could take place in section by section.
- Substring-matching for two words is optional.
- Other field-matching.
Conclusion

The intelligent broker architecture, as an add-in layer between user and search engine, performs many tasks neither user nor search engine can do. Such tasks are building a set of qualified keywords, construction of multiple queries with multiple qualified keywords, analysis leading to grouping and categorization, and retrieval of a tree of reference patents.

The JavaBeans-base intelligent broker makes itself a reusable component.

It can be extended to the other information search on Internet.