Knowledge-Pattern Based Information Extraction

Magdy Aboul-Ela
Sadat Academy for Management Sciences
Computer and Information Systems Department
P.O. Box 2222, Kournich-ElNile, El-Maadi, Cairo, Egypt

ABSTRACT: - Information Extraction (IE) is a technology for "reading" reports and picking out the bits of information that are needed by users. Hypermedia relies on a combination of knowledge representation e.g. semantic links, text analysis; and "canned" knowledge in different presentation formats. Successful information access and presentation depends on an information base where the information is represented, and not only contains a presentation of the knowledge. A weak representation of the knowledge, and the limitations of the knowledge, lead to difficulties in finding the relevant information, and may also cause the system to retrieve information that is incorrect in the current context, but would be correct in another usage situation. Combining knowledge structures, with "canned" knowledge provides better knowledge structure. This paper presents a proposed framework of how to check and compare the extracted information from different documents and to what extent they are relevant to the user profile. The framework is based on the pattern similarity between the indices of the knowledge representation structure, or the knowledge-patterns, which includes all the elements of the knowledge structure: *entities, attributes, actions, scripts, relations, and operators or rules*, in addition to the relevant linguistic rules.

KEYWORDS: - Information Extraction, Pattern Similarity, Case Based Reasoning, And Natural Language Processing.

1. Introduction

As the amount of available information in e-document is dramatically increasing, the ability for rapid and effective access to information is becoming critical. This has been known as the knowledge extraction problem for hypermedia. What information to retrieve, and how to present it, are not only dependent on the information content itself, but also on the user's profile and the usage context. However, successful information access and presentation depends on an information base where the information is represented, and not only contains a presentation of the knowledge [3]. A weak representation of the knowledge, and the limitations of the knowledge, do not only lead to difficulties in finding the relevant information, but may also cause the system to retrieve information that is incorrect in the current context (but would be correct in another usage situation) [2].

The main facilities for a user to search within the enormous amount of information available in the Web are the so called search engines. In order to access and classify information contained in Web sites concerning a specific domain of interest, one needs to represent the domain, the structure of the generic site and of the pages, and the terminology about the domain. There are many formalisms one can choose: simple formalisms are easy to process automatically but difficult to interpret (e.g. feature vectors), whereas more complex ones are difficult to use but may allow for an automatic interpretation (e.g. kwowledge representation systems) [4].

Traditional systems for information modeling present some limitation: Entity-Relationship (ER) model are not suited to represent typical hypertext structures, while Object-Oriented (OO) model are more feasible, but still lack the flexibility needed to handle the variety of structures that one can find in the Web. Also, Description Logics (DL) as a representation formalism. DL can be used as a modeling language, because of the close relationship with semantic data models, and also offer reasoning facilities to automatically classify concepts (i.e. entities).

Information Extraction [IE] is a technology for "reading" reports and picking out the bits of information that are needed by users. If you have a number of articles on mergers and acquisitions, then you can see a pattern emerging in the kinds of bits of information that would normally be extracted. This can be used to define a "template" - a table with slots that can be instantiated with the bits of information that can be extracted from a given article [8]. The template therefore lists the things we are interested in, though a given article does not necessarily instantiate every slot in the template [3].

2. Language Analysis for IE

While there has been a great deal of work on extracting data from databases, the majority of data within most businesses (and the data available about their competitors) is not in databases, but in material written in human languages, such as reports, brochures, manuals, etc. For this material to be effectively used, intelligent language analysis is necessary. Before knowledge can be acquired, formalized and refined, it naturally has to be extracted, or retrieved, from the body of text in which it is embedded. When the body of text in question is electronic, then corpus analysis tools come into play to help researchers derive meaningful data from their corpora. The Text Analyzer (TA) component should have the ability to extract sentences showing the semantic relations that hold between concepts, thereby having the potential to help semi-automate this kind of knowledge extraction [2].

Text Analyzer

The Text Analyzer (TA) is a type of corpus-analysis tool that enables users to extract and analyze certain kinds of information contained in electronic documents. The TA is proposed as a tool for any person or group of people "whose job requires them to search for knowledge in documents". The program's developers name specifically, among others, terminologists as a group of people who would benefit from this technology. Traditionally, terminologists examine vast amounts of text (a process called scanning), looking for terms and discovering the conceptual network of a given subject field. Their job would be greatly facilitated in terms of decreased time and increased productivity if the scanning could be at least semi-automated.

The TA has a number of operations, the main operations are the following [4]: ***Preprocessing* (**sentence delimiting, part of speech tagging, finding and grouping compound nouns) and ***Main Processing* (**frequency operations, concordance, collocations, conceptual operations).

Linguistic patterns

If we consider the linguistic structures expressing semantic relations to be devices, we realize that they can be very useful tools for knowledge extraction from texts. *How does information extraction work?* Suppose we are handling news reports on mergers and acquisitions [3]. One of the obvious starting points for processing it is to go through it looking for proper nouns. By pattern matching with an appropriate lexicon, people's names, geographical names, and most importantly, company names can be identified. Similarly, dates and financial values can be easily picked out. Once this information has been picked out, some structuring is called for to help determine the overall meaning of the text. In contrast to say information retrieval, the use of very common words such as "the”,” of", and "from" can be very important in determining the meaning of phrases. Other kinds of ambiguity surround issues of co-ordination in a sentence and between sentences. For example, *and* can connect a wide variety of phrases, and deciding what it is connecting can be difficult to determine. However, syntactic and simple semantic rules are not always sufficient to resolve ambiguities, and deeper domain knowledge is required. Use of **semantic knowledge** becomes even more important when taking the ***parsed text*** and trying to complete the **template**. The sense of the two occurrences of the verb maybe similar but should result in quite different instantiated templates. Part of the process of deciding which templates to complete depends on resolving the ambiguity surrounding words such as the main verb.

Integrating all the knowledge sources, *semantic**,** Pragmatic**,** and Syntactic**,*** is used to comprehend the text, and can be represented in the following knowledge representation structure [5], [6]:

Entity/Action {
  Entity-Name, Relation,
 Script of action, Attribute [1..n]}.
Relation {
  Relation-Type [1..n], Entity[1..n]).
Script of action (
   Script-name, Roles: entity [1..n],
   People: entity [1..n], Initial-State: state,

Goal-State: state, Events-Scenario:Action [1..n]}.
Attribute { isa: Attribute-Class,
Attribute-Name, Attribute-Value [1..n]}.
Attribute-Class {
Domain: string/number/entity, Range: range-values, Range-Constraint: not range-values}.
Attribute-Value {
Operation: [arithmetic, logical],
Values: [entity,string], RelationTo}.
RelationTo {
Relation, Object}.
State/Object { isa: Entity, *Id.}.*

This knowledge representation structure can adopt any text structure, such as:

```
Event: (main verb, or infinitive): ...
Type of Event: e.g. Action
Agent: ...          Object1: ...
Time: ...           Location: ...
Condition: ...      Exception: ...
Reason: ...         Recipient: ...
Behavior: ...       Beneficiary: ...
Instrument: ...     Topic: ...
Focus: ...
```

Where ***Topic*** represents what does the text describe? And ***Focus***, repesents the the most impotant information relative to the context. The Topic and Focus can be defined accoding to pragmatic rules, for example:

```
If event is not action
then Topic = event and
      Focus is Object1.
If event is action and
   Agent is not null and
   Object is null and
   reason is not null
then Topic is reason and Focus is
      Agent.
If event is action and
   Agent is not null and
   Object is not null
then Topic is (Action and object) and
      Focus is Agent.
```

The Knowledge index, pattern, of the above knowledge representation is [5]:

```
Knowledge Pattern
{
Entities(
  Entity-1( (Attributes –1,…,Attribute –n),
            (Operator-1,…, Operator-n))
  Entity-2(… ),…,Entity-n(… ))
Attributes(Attribute-1
            (Entity-1,…, Entity-n),
          Attribute-2(),…,Attribute-n(…))
Scripts(Script-1(
            Goals(Goal-1, …, Goal-n)
            Props(Prop-1…,  Prop -n)
            Roles(Role-1,…, Role –n)
            Conditions(
              Condition-1,…,Condition-n)
            Actions(action-1,…,action-n)
          Script-2(…), …Script-n(…))
Operators(Operator-1
                (Action-1(Entity-1),
                   …        …
                 Action-n(Entity-n)),
          Operator-2(…),…,Operator-n.(…))
Actions(Action-1(attribute-1,…, attribute-n),
        Action-2(…),…,Action-n(…))
Relations(Relation-1(Entity-i1,…, Entity-j1),
          Relation-2(…),…,Relation-n(…))
Events(Event-i1(…),Event-i2(…),…,Event-in(…))
}.
```

3. Knowledge-Pattern based Information Extraction

The pattern-based extraction method extracts information based on the above knowledge Pattern , which have the same structure in the users profile. Consider the pattern as *object o* is a list of continuous fields representing a piece of knowledge. Furthermore, a *template* is a specific object indicated by the users. The similarity between the template and potential objects is known as *pattern similarity*.

Knowledge-Pattern Similarity

Pattern similarity measures how much two objects match with each other. The concept of *matched fields:*

> *A field f_i matches with another field f_j, denoted by $f_i = f_j$, if both fi and fj have equal values.*

The **rules of matching**, based on linguistic analysis first and are defined in a knowledge base before deciding whether to proceed with measuring the degree of matching or not, For example:

```
if events have the same meaning
  and corresponding fields have the
  same meaning
then     patterns are matched
           else
if TOPICS are equal and FOCUS are
   equal
then     patterns are matched
else
if TOPICS are equal and FOCUS are
   not equal
then     patterns are not matched
           (but related)
else
if TOPICS are not equal
then     patterns are not equal.
```

(but may relate)

Due to the complexity of linguistic analysis, and to reduce the ambiguity, **other rules** for ***pattern similarity measure*** between two objects are integrated, and based on to the followings definitions [3]:

- *An object* p=*(p1, p2, ..., pm) is a* sub-object *of* q=*(q1, q2, ..., qn) if there is a* sublist *of* q, *(qi1, qi2, ..., qim) such that for each k, 1 < k < m-1 => ik < ik+1 and 1 < k < m => pk = qik.*

- *An object* p=*(p1, p2, ..., pm)* matches *with the object* q=*(q1, q2, ..., qn), denoted by* p= q, *if m = n and for each k, 1 < k < m => pk = qk.*

- *For objects* p *and* q, *the* pattern similarity measure, *denoted by* PSM, *is the* maximum size *of the* sub-objects *pi and qj such that pi = qj, and the pattern similarity score PSS, which is the* ratio *of* PSM *to* the average size *[3]*:

  $PSS(p, q) = (PSM(p, q) /((size(p) + size(q))/2)) *100;$

  For example, for objects *p, q*:

  p=(Account-No., :, (, 518,), 345, -, 9465) , and
  q=(Card-No., :, 1, -, 518, -, 345, -, 9468),

  The *maximum matched sub-objects* are:
  (:, 518, 345, -, 9465) = (:, 1, 518, -, 345).

  Thus,
  PSM(*p*, *q*)=5, and
  $PSS(p, q)=(5/((8+9)/2))*100 = 58\%;$

In order to reduce the effect of *false matches* between large objects, at first the *ratio* of *PSM* to *the average size* of the objects is considered to reflect the real *pattern similarity*. However, for two objects whose *average size* is 10 and *PSM* is 5, the *ratio* is 50% that is the same as that of objects whose *average size* is 2 and *PSM* is 1. In practice, the former should have a higher *similarity* since the one *matched field* in the latter may be a *false match*. Therefore, we multiply the *ratio* by the *PSM* to reflect it. The final result for the *pattern similarity score PSS* [3].

$$PSS(p, q) = (PSM(p, q) / ((size(p)+size(q))/2)) *PSM(p, q)*100;$$

In the above example,
$PSS(p, q)=(5/((8+9)/2))*5*100= 290;$

If we assume the object *p* is the *sample*, then the *PSS* is enough to distinguish the object *q* from other parts. However, if there is another object *l,* it is hard to determine which (*q* or *l*) should match with *p* since *PSS*(*p*, *q*)= *PSS*(*p*, *l*).

But in certain patterns matching some fields may have higher affects in similarities measure between patterns as a whole, than others; even in the case of *PSM* has a small value. Instead of using the *ratio* of *PSM* to *the average size* of the objects, the ratio of the summation of the fields' weights multiplied by the similarity function, to the total weights of all fields is used, where the weight is the measure of the importance of the feature, and has a default value 1 for all features, and then increase with an order of magnitude according to its importance in the object or pattern, e.g. the features in the knowledge pattern included in the fields, such as, entity, attributes, relation may have the weight 3, but the weight of the feature defined as the focus feature is 5, and for the topic feature is 7. The feature means here the subfield in the knowledge pattern, which has a value.

$$PSS(p, q) = (\Sigma(w_i * Simf_i(f_p , f_q))/((\Sigma w_p + \Sigma w_q)/2))*(PSM(p, q)*100);$$

Where:
w_i , w_p , w_q are the weights or the importance of the field f_i ., f_p, f_q in pattern.
$Simf_i(f_p , f_q)$ is the similarity function between the matched fields f_p , f_q at objects p,q.
$Simf_i(f_p , f_q) = 1 – (|f_p - f_q|/ \max(|f_p |, | f_q|))$ if the values of the fields are numbers.
$Simf_i(f_p , f_q) = 1$ if the values are symbols and equals.

If the values are symbols but not equal, the meaning must be represented in a semantic net for each value using a semantic lexicon [9], and then make the graph matching similarity measure *GMS*.

$$\text{GMS}(g_p , g_q) = 1 - (|\text{mcs}(g_p , g_q)| / \max(|g_p |, | g_q|))$$

Where mcs (g_p , g_q) is the maximum common subgraph of two graphs g_p , g_q and $|g_p|$, $|g_q|$ is the number of nodes of graph g_p , g_q [7].

There is a threshold value for similarity measure, and must be adapted for the different fields until reaching the good result, which depends on the user requirements for the degree of accuracy.

The Algorithm

Knowledge-Pattern similarity can be used to extract the desired information from unstructured or structured text based on the sample specified by the users. The following is the proposed algorithm:

- Paraphrase the text into declarative sentences to start Knowledge acquisition process for creating the knowledge structure and its knowledge index or pattern as in the framework for open mind learner [5].
- Apply the proposed linguistic rules to determine the focus and the topics of the document and user profile or the template.
- Determine the weights of each feature according to the proposed predefined rules.
- Calculate PSS for the two knowledge patterns, using similarity function and GMS function.

Example:

Consider the following two pieces of knowledge:

> *P:- Text mining is the process of extracting the patterns from text.*
> *Q:- Data mining: is a process of extracting information from database.*

After processing these sentences using the linguistic rules and semantic lexicon [5], [6]; the following knowledge Structures are created:

Knowledge structures:

```
P:- Entity: Text mining
  { relation
  {   relation-name = "isa",
      relation-type = "inheritance"
      entity[1]: process
       {relation{ relation-name = "of",
        relation-type ="association"
                      action[1]: extracting
                     {attribute:{attribute-name: object
                       attribute-value:
                       {entity: pattern
                        {attribute-name: location
                          attribute-value:
                          {entity: text}}}}}}}}.
Q:- Entity: Data mining
  { relation
  {   relation-name = "isa",
      relation-type = "inheritance"
      entity[1]: process
       {relation{ relation-name = "of",
        relation-type ="association"
                      action[1]: extracting
                     {attribute:{attribute-name: object
                       attribute-value:
                      {entity: pattern
                       {attribute-name: location
                         attribute-value:
                       {entity: database}}}}}}}}.
```

The knowledge patterns for the above two structures are:

P: - *Entities (Text Mining, process, pattern, text), Attributes(object (extracting, pattern), location (pattern, text)), Actions(extracting(pattern)), Relations(isa(text mining, process), of(process, extracting)).*

Q: - *Entities (Data Mining, process, Information, database), Attributes(object (extracting, information), location (information, database)), Actions(extracting(information)), Relations(isa(data mining, process), of(process, extracting)).*

The similarity functions will have the values in the following table:

i	f_p , f_q	w_i	Simf_i	w_i* Simf_i
1	*Text mining,* *Data mining*	*3*	*1/3*	*1*
2	*Process,* *Process*	*3*	*1*	*3*
3	*Pattern,* *Information*	*7*	*1/2*	*7/2*
4	*Text,* *database*	*3*	*1/4*	*3/4*
5	*object (extracting, pattern),* *object (extracting, information)*	*3*	*1/2*	*3/2*
6	*location (pattern, text)* *location(Information,database)*	*3*	*1/3*	*1*
7	*Actions(extracting(pattern)),* *Actions(extracting(Information))*	*5*	*1/2*	*5/2*
8	*isa(text mining, process),* *isa(data mining, process)*	*3*	*1/3*	*1*
9	*of(process, extracting),* *of(process, extracting)*	*3*	*1*	*3*

Where the weight of topic is 5, and the weight of focus is 7 and the other fields are 3.

From the above example rules:
Topic1: Action = extracting, Focus1: object= pattern; and Topic2: Action = extracting
Focus2: object= Information

Where $Simf_i$ is calculated according to the graph matching similarity *GMS* as defined in the semantic lexicon. Consider the case of *data* and *text* words, where *text* is defined as sequences of words and include *data* which is defined as attribute-value pair nods, the ratio of the number of matching nodes is 1/3. Therefore PSS is calculated as follows:

$PSS(p, q) = (\sum(w_{i*} \mathrm{Simf}_i(f_p , f_q))/((\sum w_p + \sum w_q)/2))*(PSM(p, q)*100) = ((69/4)/33)*(9*100) = 470.45.$
Where *PSS* in the case of the complete similarity is 900.

4. Conclusions

We proposed in this paper a framework of information extraction from text document. We are trying by this framework to complement and increase the efficiency of information extraction techniques, by applying similarity measure technique on the knowledge patterns, which is the knowledge index to the knowledge structure. The threshold value of the degree of the similarity with the predefined rules for the weights of features must be adapted on many iterations and different cases. The similarity function is used in case of the feature value is numbers, and when the attribute value is symbol the graph matching similarity measure is applied to determine to what extent the difference between the two features. The proposed framework consists mainly of linguistic knowledge base, includes the semantic and pragmatic rules, and semantic lexicon; the rules of assigning the importance of the features, or weights; the knowledge pattern that map a description and definition of the elements of a piece of knowledge represented in the knowledge structure; and the procedure of measuring the pattern similarity score. The framework is independent on the format of the documents. This knowledge pattern, or index, has an advantage of accessing any element of knowledge, and making the matching from different views: *entities*, *relations, actions, attributes, or rules,* based on the meaning of the text within its context, in addition the training method can be incorporated to adjust the weights of measurement. The linguistic rules is updated according the language used, and any new knowledge can be added to the current structure.

References:

[1] Adrienne Franco, M.L.S., M.S. and Richard L. Palladino ,M.L.S., M.S. ; Finding Quality Information on the World Wide Web, Tenth Annual Conference of the International Information Management Association, Iona College Libraries, New Rochelle, NY , October 14, 1999

[2] S. Soderland , Learning to extract text-based information from the world wide web, In Proceedings of 3rd International Conf. on Knowledge Discovery and Data Mining (KDD-97), 1997, 251-254.

[3] Mattia De Rosa, Luca Iocchi, Daniele Nardi; “Knowledge representation techniques for information extraction on the Web”, Nikos Drakos, Computer Based Learning Unit, University of Leeds ,1998.

[4] Laura Davidson, Knowledge Extraction Technology for Terminology, University of Ottawa, Ottawa, Ontario, Canada, 1997.

[5] Aboul-Ela, Magdy; “Framework For Open Mind Learner”, The Egyptian Computer Journal, Institute of Statistical Studies and Research, Cairo University, December 2004.

[6] Aboul-Ela, Magdy; “ A Framework for an Intelligent Problem Solver”, WSEAS (The World Scientific and Engineering Academy and Society) Transaction Journal on Computers, Issue 5, Volume 3, November 2004, Page 1563.

[7] Bunke, H. and Shearer, K. “A Graph Distance Metric Based on Maximal Common Subgraph, Pattern Recognition Letter, Vol 19, Nos. 3-4 1998, pp. 255-259.

[8] Verspoor, M, C.; Papcun, J, G. and Sentz, K. “A Theoretical Motivation for Patterns in Information Extraction”, Los Alamos Unclassified Report LAUR 02-1504, 2003.

[9] Gheith, Mervat; Aboul-Ela, Magdy; and Arafa, Waleed; “Lexical Acquisition for Information Extraction from Arabic Text Document”, WSEAS (The World Scientific and Engineering Academy and Society) Transaction Journal on Systems, July 2004.