Natural Language Information Interpretation Representation System: Machine Learning Based Approach

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 12, Issue No. 1, 42–52

https://journals.mirlabs.in/index.php/ijcsnt
DOI-10.18486/ijcsnt/12.1.159
IJCSNT

Banerjee Partha Sarathy[1], Chakraborty Baisakhi[2], Banerjee Jaya[3], Anand Utkarsh[4]

Abstract

The two synonymous terms: Text Mining (TM) and Information Extraction (IE) are very closely knit. Text Mining is all about tracing patterns in the natural language text where as Information Extraction is extraction of a specific information in the unstructured data. IE modules have off late been automated using the Machine Learning approach. Here the system integrates the TM and IE process in the light of Machine Learning to make a better knowledge extraction system for information interpretation and representation of unstructured data module NLIIRS. This paper presents a framework by the application of a machine learned information extraction system on text to trace interesting relations. The rules mined while developing a database from the text can be used in future documents for efficient information extraction as the recall value improves.

Keywords

Text Mining (TM), Information Extraction (IE), NLIIRS, Machine Learning

Received: 22 December 2022; **Revised:** 16 April 2023; **Accepted:** 25 April 2023;
Published: 30 April 2023;

[1]Research Scholar, [2]Asst. Professor, [3]M.Tech CSE, [4]BTech CSE
[1, 2, 3, 4]Department of Computer Science & Engineering,
[1, 2, 3]National Institute of Technology Durgapur, West Bengal, India,
[4]Jaypee University of Engineering & Technology, Guna, Madhya Pradesh, India.

Corresponding author:
Email: psbanerjee.tec@nitdgp.ac.in

Introduction

The prevalence of structured databases like the the RDBMS has now been gradually out shadowed due to the emergence of a huge amount of unstructured data. The unstructured data is mostly available in the form of natural language text. Hence over the years there has been a continuous shift of attention from structured data handling to unstructured data mining and management as in [4,18,19,21,22,27]. This paper address to a novel approach of information mining by capturing the advantages of machine learning and IE and TM. Knowledge discovery from databases KDD and datamining also plays an equally important role. In a nutshell we may assume that this work integrates the benefits of KDD, IE and machine learning. The task of information extraction is difficult but with the introduction of machine learning it has been optimized to a certain extent. By using machine learning the excerpt may be labeled and annotated but this annotated text may be altered manually so that the hybrid model can be more efficient. This annotated segment of natural language text by two methods: machine learning and manual may then be applied over newer and larger segment of textual information to construce the database. However this system has its own limitations due to its tedious nature and hence it is error prone. Due to this fact the reliability of this system has to be monitored. A way out to this problem is taking the benefit of KDD and applying it to the information extraction module. For example we may see the the information technology sector jobs requiring the cassendra skills are the database administrator jobs in most of the situations. Now at this point if the information extraction module is able to locate cassendra but unable to locate the database administrator then we may say that there is discrepancy in the information that has been extracted.

Related work

In the initial years text mining used to be the only method of information retrieval. Earlier knowledge extraction module assumed that the information to be extracted from the data provided is already in the structured form, where as in contrast to this most of the widely available information is in the form of natural language text. The information extraction process the data so as to transform it into a semistructured form. The development of an IE system is a tedious task but over the years this process has been automated using the modernized machine learning tools as described in [5,7,9,23]. By automating an IE system the possibility of addition of Noise in the information increases considerabley as in [3,8,12]. To avoid this problem very less amount of work has been done. The earlier knowledge extraction system as been shown in figure 1. There is a solution to this problem which we will discuss in detail in later stages like, instead of fully getting dependent on the system for correctly extracting the information we may at first make a small set of data manually trained by creating the knowledge base manually [29,30]. This very trained dataset about whch description is provided in [14,16,17] can then be machine trained so that this same rule may now be applied over a larger segment of unstructured information from which the information is to be extracted correctly.

As described in [20] TM initially automated the process of information extraction. There are several techniques like association rule mining and episode rule mining but all these techniques and many others have been used very frequently and with very little technical difference between them as described in [2]. The usefulness of this modle can be well understood from the information provided in [5,9,23] like in web pages, city tour guides and in information announcement.

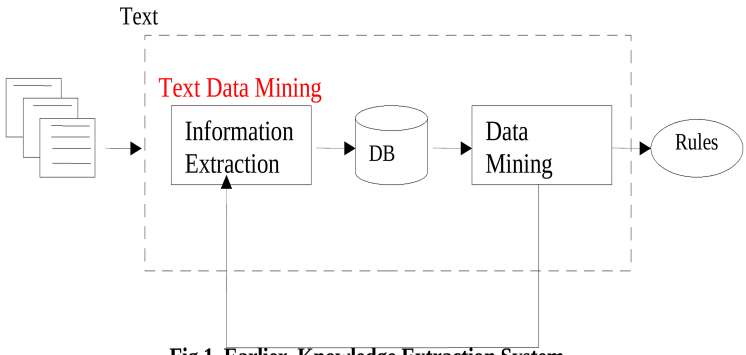

Figure 1. Earlier Knowledge Extraction System

Proposed System:

The basic idea to develop this module can get clear from this very example. As an example, for a job search portal if there is an information that expertise in MongoDB skills is required and this information is almost similar to the NoSQL skilled jobs in most of the cases. It may happen that a system may be able to figure out MongoDB in the language section but failed to find the class of that data[31,32]. We have developed machine learning techniques to automatically construct information extractors for job postings. By retrieving the information from a natural language text of job postings, a structured, searchable database of jobs can be automatically constructed, thus making the data in online text more easily accessible[33,34]. IE has been shown to be useful in a variety of other applications, e.g. seminar announcements, restaurant guides, university web pages, apartment rental ads, and news articles on corporate acquisitions. We consider the task of first constructing a database by applying a learned information-extraction system to a corpus of natural-language documents. Then, we apply standard data-mining techniques to the extracted data, discovering knowledge that can be used for many tasks, including improving the accuracy of information extraction.

NoSQL jobs. Under these circumstances we may say that there is an information extraction anomaly. In general recall of any system is mostly lower than that of its precision, that means the accuracy of information correctly extracted is much lower than that of the information extracted is correct. As described in[13]. This paper address this very fact that the recall part may be considerably improved by adding some rules for extraction which may be appended from time to time and hence training the machine. We have used the technique used in[28] for the data availability.An application of[1,10] is used for inferring rules from the available trained data set. The filtering of the unimportant document id done using the text categorizer as in[26]. We consider the task of first constructing a database by applying a learned information-extraction system to a corpus of natural-language documents. Then, we apply standard data-mining techniques to the extracted data, discovering knowledge that can be used for many tasks, including improving the accuracy of information extraction.

After constructing an IE system that extracts the desired set of slots for a given application, a database can be constructed from a corpus of texts by applying the IE extraction patterns to each document to

create a collection of structured records. Standard KDD techniques can then be applied to the resulting database to discover interesting relationships. Specifically, we induce rules for predicting each piece of information in each database field given all other information in a record. In order to discover prediction rules, we treat each slot-value pair in the extracted database as a distinct binary feature

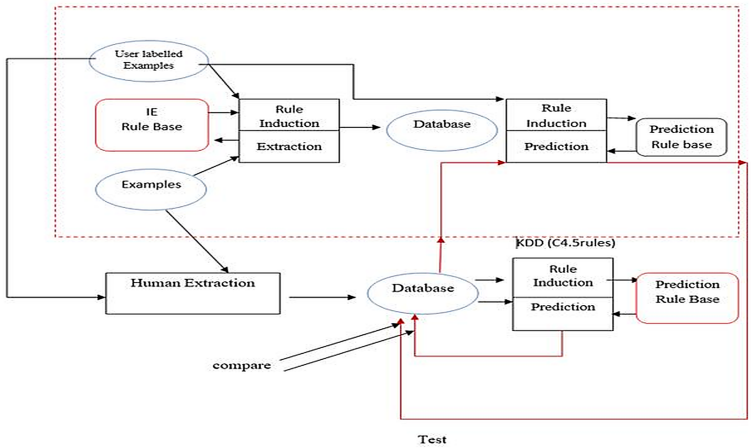

Figure 2. Proposed System

This section will illustrate the details about the proposed module, machine learning based NLIIRS for knowledge extraction from unstructured data. The first step in this module is to develop the database by implementing a learned information extraction rule on the available text in the form of natural language. We then apply our information extraction algorithms that may be used for improving the extraction process. The model at first segments the document provided and hence trains itself which can now be tested over novel set of natural language text information as described in [6,15].

Natural Language Text Document

Heading: Software Designer

Area: New Delhi, Kolkata

The applicant will be answerable for the development and execution of SDLC and a successful candidate mush have an experience on these:

One or more of: NoSQL, Python

Programming in C#, C

Database access and integration: JDBC

CGI and scripting: one or more of JSP,.NET

Exposure to the following is a plus: WebHosting, Backend

A BTech and 4+ years experience (or equivalent) is required.

Information Categorised as:

Heading: “Software Designer”

Area: “New Delhi, Kolkata”

languages: “C#”, “C”

platforms: “NoSQL”, “Python”
applications: “JDBC”
Expertise: “NoSQL Database”
Qualificaton: “BTech”
years of experience: “4+ years”

Fig 3 A Natural Language Text Sample and the Categorised Information

The rule learning needs to have a trade off between the precision and the recall as discussed in[11]. In the below mentioned table the system shows the official as well as its similar forms which is supposed to be retrieved when the system goes for categorization.

Table 1. Similar Form Dictionary

Official Notation	Similar Forms
“Access”	“MS Access”,“ Microsoft Access”
“ActiveX”	“ActiveX”
“AI”	“Artificial Intelligence”
“Animation”	“GIF Animation”, “GIF Optimization/ Animation”
“Assembly”	“Assembler”
“ATM”	“ATM Svcs”
“C”	“ProC”,“Objective C”
“C++”	“C ++”,“C+ +”
“Client/Server”	“Client Server”, “Client-Server”, “Client / Server”
“Cobol”	“Cobol II”, “Cobol/400”, “Microfocus Cobol”
...	...

In figure 4 as well as in figure 5 we have listed some of the production rules for Advertisement for Employment as well as Bio Data Advertisements. The production rules may be understood by the following examples: lets say when oracle is categorized as an application QA Partner is also categorized as application then if SQL is figured in the corpus then it will be automatically classified as an application. This kind of classification is for homogeneous categories like the application.

Advertisement for Employment with Production Rules

- Oracle € *application* and QA Partner € *application* →SQL € *application*
- HTML € *language* and WindowsNT € *platform* and Active Server Pages € *application* → Database € *area*
- Java € *language* and ActiveX € *area* and Graphics € *area* →Web € *area*
- UNIX € *platform* and Windows € *platform* and Games € *area* →3D € *area*
- AIX € *platform* and Sybase € *application* and DB2 € *application* →Lotus Notes € *application*
- C++ € *language* and C € *language* and CORBA € *application* and Title=Software Engineer →Windows € *application*

Figure 4: Production Rules for Employment Oppertunities

Bio Data Advertisements with Production Rules

- HTML € *language* and DHTML € *language* →XML € *language*
- Illustrator € *application* → Flash € *application*
- Dreamweaver 4 € *application* and Web Design € *area* → Photoshop 6 € *application*
- MS Excel € *application* ⇒ MS Access € *application*
- ODBC € *application* ⇒ JSP € *language*
- Perl € *language* and HTML € ⇒ Linux € *platform*

Figure 5: Production Rules for Bio Data Advertisements

The Algorithm

The algorithms for segmenting the category and call of the information manually or by a training algorithms is provided in figure 6 and the process of extending the database created during the training phase by implementing the production rule is provided in figure 7. The algorithms are self explanatory in nature and the first algorithm's output acts as an assisted input to the second algorithm.

Input: T is the set of unstructured information in the form of document.
Output: PR is the set of projected extraction rules.
Function RuleMining (T)
Determine Z, a threshold value for rules to be valid
A database is created of the pre assigned categories in the trained set (done by using IE to the Natural Lang Text, T)
For each pre assigned tag on T € T **do**
  M: = total categories of T
  Then transform M to binary features
A Production Rule Dictionary is created which is PR
For every Production Rule R € PR do
  Validate *R* all set of forthcoming data
  If the percentage of correctness of *R* is lower than Z
    Delete *R* from *PR*
Return *PR*.

Algorithm Extracted Categories

Input: PR is the Production Rule in the created Dictionary
  T is the set of unstructured information in the form of document.
Output: M is the total categories of T determined.
Function Knowledge Retrieved (PR,*T*)
M: = ∅
For each example T € T **do**
  Extract categories from T using Production Rule and append them to M
  For each rule *R* in the Production Rule base PR **do**
    If *R* fires on the current extracted fillers

If the predicted categories is a part of T
Extract the category and add it to M
Return *M*

Knowledge Extraction Algorithm

To summarize, documents which the user has annotated with extracted information, as well as unsupervised data which has been processed by the initial IE system are all used to create a database. The rule miner then processes this database to construct a knowledge base of rules for predicting slot values. These prediction rules are then used during testing to improve the recall of the existing IE system by proposing additional slot fillers whose presence in the document are confirmed before adding them to final extraction template.

Results

We calculated the precision, recall and f measure of the data for around 500 postings of employment opportunities and of similar number of bio data. The data is trained and segregated and then the values of precision and recall are computed.

$$precision = \frac{\textbf{\textit{Total number of categories correctly predicted}}}{\textbf{\textit{Number of categories predicted to be available}}} \quad (1)$$

$$recall = \frac{\textbf{\textit{Total number of categories correctly predicted}}}{\textbf{\textit{Number of actual categories}}} \quad (2)$$

Table 2. Performance results for categorized as well al unclassified information

Total data sets for Prediction Rules	Precision Value	Recall Value	F-Measure
0	97.4	77.6	86.4
540(Categorized)	95.8	80.2	87.3
540+1000(Unclassified)	94.8	81.5	87.6
540+2000(Unclassified)	94.5	81.8	87.7
540+3000(Unclassified)	94.2	82.4	87.9
540+4000(Unclassified)	93.5	83.3	88.1
Matching Values	59.4	94.9	73.1

Evaluation

Discovered knowledge is only useful and informative if it is accurate. Therefore it is important to measure the accuracy of discovered knowledge on independent test data. The primary question we address in the experiments of this section is whether knowledge discovered from automatically extracted data (which may be quite noisy due to extraction errors) is relatively reliable compared to knowledge discovered from a manually constructed database. While testing the system around 600 manualled annotated openings to the jobs were collected. A series of training and test sets were generated and wre cross validated. With

this data there was and an extra 4000 unlabelled textual documents to be later on used as inputs for the text miner. The rules were generated for the same and for filling the slots under the various category for that particular case like the job vacancy case which are already described in the algorithm.

It has been clearly stated that the values that how the precision, recall and F-measure changes when extra unannotated documents are added to have a extended data repository preceding to extracting and applying the prediction rules. The 540 labeled examples used to train the extractor were always provided to the rule miner, while the number of additional unsupervised examples were varied from 0 to 4,000. The results show that the more unsupervised data supplied for building the prediction rule base, the higher the recall and the overall F-measure. Although precision does suffer, the decrease is not as large as the increase in recall. Although adding information extracted from unlabeled documents to the database may result in a larger database and therefore more good prediction rules, it may also result in noise in the database due to extraction errors and consequently cause some inaccurate prediction rules to be discovered as well.

Future work

Good metrics for evaluating the interestingness of text-mined rules are also needed. One idea is to use a hierarchical lexical network to measure the semantic distance between the words in a rule, preferring "unexpected" rules where this distance is larger. WordNet is general purpose and does not capture many of the semantic similarities in particular domains. Using a domain-specific taxonomy would be helpful for finding interesting rules.

Conclusions

In this paper, we have presented an approach that uses an automatically learned IE system to extract a structured databases from a text corpus, and then mines this database with existing KDD tools. Our preliminary experimental results demonstrate that information extraction and data mining can be integrated for the mutual benefit of both tasks. IE enables the application of KDD to unstructured text corpora and KDD can discover predictive rules useful for improving IE performance. This paper has presented initial results on integrating IE and KDD that demonstrate both of these advantages. Text mining is a relatively new research area at the intersection of natural-language processing, machine learning, data mining, and information retrieval. By appropriately integrating techniques from each of these disciplines, useful new methods for discovering knowledge from large text corpora can be developed.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB-94), pages 487–499, Santiago, Chile, Sept. 1994.

[2] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. ACM Press, New York, 1999.

[3] S. Basu, R. J. Mooney, K. V. Pasupuleti, and J. Ghosh. Evaluating the novelty of text-mined rules using lexical knowledge. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), pages 233–239, San Francisco, CA, 2011.

[4] M. W. Berry, editor. Proceedings of the Third SIAM International Conference on Data Mining(SDM-2003) Workshop on Text Mining, San Francisco, CA, May 2013.

[5] M. E. Califf, editor. Papers from the Sixteenth National Conference on Artificial Intelligence (AAAI-99) Workshop on Machine Learning for Information Extraction, Orlando, FL, 2010. AAAI Press.

[6] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), pages 328–334, Orlando, FL, July 2011.

[7] C. Cardie. Empirical methods in information extraction. AI Magazine, 18(4):65–79, 2014.

[8] C. Cardie and R. J. Mooney. Machine learning and natural language (Introduction to special issue on natural language learning). Machine Learning, 34:5–9, 2016.

[9] F. Ciravegna and N. Kushmerick, editors. Papers from the 14th European Conference on Machine Learning(ECML-2003) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD-2003) Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, Sept. 2013.

[10] W. W. Cohen. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning (ICML-95), pages 115–123, San Francisco, CA, 2015.

[11] W. W. Cohen. Learning to classify English text with ILP methods. In L. De Raedt, editor, Advances in Inductive Logic Programming, pages 124–143. IOS Press, Amsterdam, 2016.

[12] W. W. Cohen. Improving a page classifier with anchor extraction and link analysis. In S. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information Processing Systems 15, pages 1481–1488, Cambridge, MA, 2003. MIT Press.

[13] DARPA, editor. Proceedings of the Seventh Message Understanding Evaluation and Conference (MUC-98), Fairfax, VA, Apr. 1998. Morgan Kaufmann.

[14] R. Feldman, M. Fresko, H. Hirsh, Y. Aumann, O. Liphstat, Y. Schler, and M. Rajman. Knowledge management: A text mining approach. In U. Reimer, editor, Proceedings of Second International Conference on Practical Aspects of Knowledge Management (PAKM-98), pages 9.1–9.10, Basel, Switzerland, Oct. 2010.

[15] D. Freitag and N. Kushmerick. Boosted wrapper induction. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), pages 577–583, Austin, TX, July 2012. AAAI Press / The MIT Press.

[16] R. Ghani and A. E. Fano. Using text mining to infer semantic attirbutes for retail data mining. In Proceedings of the 2012 IEEE International Conference on Data Mining (ICDM-2002), pages 195–202, Maebash City, Japan, Dec. 2016.

[17] R. Ghani, R. Jones, D. Mladenić, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the Web. In D. Mladenić, editor, Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining, pages 29–36, Boston, MA, Aug. 2000.

[18] M. Grobelnik, editor. Proceedings of IEEE International Conference on Data Mining (ICDM-2001) Workshop on Text Mining (TextDM'2001), San Jose, CA, 2001.

[19] M. Grobelnik, editor. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence(IJCAI-2003) Workshop on Text Mining and Link Analysis (TextLink-2003), Acapulco, Mexico, Aug. 2003.

[20] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, 2000.

[21] M. A. Hearst. Untangling text data mining. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), pages 3–10, College Park, MD, June 1999.

[22] M. A. Hearst. What is text mining? http://www.sims.berkeley.edu/˜heast/text-mining.html, Oct. 2003.

[23] N. Kushmerick, editor. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001) Workshop on Adaptive Text Extraction and Mining, Seattle, WA, Aug. 2011. AAAI Press.

[24] S. Loh, L. K. Wives, and J. P. M. de Oliveira. Concept-based knowledge discovery in texts extracted from the Web. SIGKDD Explorations, 2(1):29–39, July 2000.

[25] A. McCallum and D. Jensen. A note on the unification of information extraction and data mining using conditional-probability, relational models. In Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data, Acapulco, Mexico, Aug. 2013.

[26] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In Papers from the AAAI-98 Workshop on Text Categorization, pages 41–48, Madison, WI, July 2014.

[27] D. Mladenic, editor. Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining, Boston, MA, Aug. 2000.

[28] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo,CA, 2015.

[29] U. Y. Nahm and R. J. Mooney. A mutually beneficial integration of data mining and information extraction. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), pages 627–632, Austin, TX, July 2000.

[30] U. Y. Nahm and R. J. Mooney. Using information extraction to aid the discovery of prediction rules from texts. In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining, pages 51–58, Boston, MA, Aug. 2000.

[31] U. Y. Nahm and R. J. Mooney. Mining soft-matching rules from textual data. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), pages 979–984, Seattle, WA, July 2001.

[32] U. Y. Nahm and R. J. Mooney. Mining soft-matching association rules. In Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM- 2002), pages 681–683, McLean, VA, Nov. 2002.

[33] J. M. Pierre. Mining knowledge from text collections using automatically generated metadata. In D. Karagiannis and U. Reimer, editors, Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002), pages 537–548, Vienna, Austria, Dec. 2002. Springer. Lecture Notes in Computer Sicnece Vol. 2569.

[34] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo,CA, 1993.