A Frame Work for Text Mining using Learned Information Extraction System

Sathish Kuppani[1]

[1] SV University

Received: 6 December 2015 Accepted: 3 January 2016 Published: 15 January 2016

Abstract

Text mining is a very exciting research area as it tries to discover knowledge from unstructured texts. These texts can be found on a computer desktop, intranets and the internet. The aim of this paper is to give an overview of text mining in the contexts of its techniques, application domains and the most challenging issue. The Learned Information Extraction (LIE) is about locating specific items in natural-language documents. This paper presents a framework for text mining, called DTEX (Discovery Text Extraction), using a learned information extraction system to transform text into more structured data which is then mined for interesting relationships. The initial version of DTEX integrates an LIE module acquired by an LIE learning system, and a standard rule induction module. In addition, rules mined from a database extracted from a corpus of texts are used to predict additional information to extract from future documents, thereby improving the recall of the underlying extraction system. Applying these techniques best results are presented to a corpus of computer job announcement postings from an Internet newsgroup.

Index terms—

Introduction n this modern culture, text is the most common vehicle for the formal exchange of information. Although extracting useful information from texts is not an easy task, it is a need of this modern life to have a business intelligent tool which is able to extract useful information as quick as possible and at a low cost. Text mining is a new and exciting research area that tries to take the challenge and produce the intelligence tool. The tool is a text mining system which has the capability to analyse large quantities of natural language text and detects lexical and linguistic usage patterns in an attempt to extract meaningful and useful information [1]. The aim of text mining tools is to be able to answer sophisticated questions and perform text searches with an element of intelligence. Technically, text mining is the use of automated methods for exploiting the enormous amount of knowledge available in text documents. Text Mining represents a step forward from text retrieval. It is a relatively new and vibrant research area which is changing the emphasis in text-based information technologies from the level of retrieval to the level of analysis and exploration. Text mining, sometimes alternately referred to as text data mining, refers generally to the process of deriving high quality information from text. Researchers like [2], [3] and others pointed that text mining is also known as Text Data The problem of text mining, i.e. discovering useful knowledge from unstructured or semi-structured text, is attracting increasing attention [4,18,19,21,22,27]. This paper suggests a new framework for text mining based on the integration of Learned Information Extraction (LLIE) and Knowledge Discovery from Databases (KDD), a.k.a. data mining. KDD and LIE are both topics of significant recent interest. KDD considers the application of statistical and machine-learning methods to discover novel relationships in large relational databases. LIE concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from free text. However, there has been little if any research exploring the interaction between these two important areas. In this paper, we explore the mutual benefit that the integration of LLIE and KDD for text mining can provide. Traditional data mining assumes that the information to be ”mined” is already in the form of a relational database. Unfortunately, for many

applications, electronic information is only available in the form of free natural-language documents rather than structured databases. Since LLIE addresses the problem of transforming a corpus of textual documents into a more structured database, the database constructed by an LLIE module can be provided to the KDD module for further mining of knowledge as illustrated in Figure 1. Information extraction can play an obvious role in text mining as illustrated. The constructing an LIE system is a difficult task, there has been significant recent progress in using machine learning methods to help automate the construction of LIE systems [5,7,9,23]. By manually annotating a small number of documents with the information to be extracted, a reasonably accurate LIE system can be induced from this labelled corpus and then applied to a large corpus of text to construct a database. However, the accuracy of current LIE systems is limited and therefore an automatically extracted database will inevitably contain significant numbers of errors. An important question is whether the knowledge discovered from this "noisy" database is significantly less reliable than knowledge discovered from a cleaner database. This paper presents experiments showing that rules discovered from an automatically extracted database are close in accuracy to that discovered from a manually constructed database.

A less obvious interaction is the benefit that KDD can in turn provide to LIE. The predictive relationships between different slot fillers discovered by KDD can provide additional clues about what information should be extracted from a document. For example, suppose we discovered that computerscience jobs requiring "My SQL" skills are "database" jobs in many cases. If the LIE system manages to locate "My SQL" in the language slot but failed to extract "database" in the area slot, we may want to assume there was an extraction error. Since typically the recall (percentage of correct slot fillers extracted) of an LIE system is significantly lower than its precision (percentage of extracted slot fillers which are correct) [13], such predictive relationships might be productively used to improve recall by suggesting additional information to extract. This paper reports experiments in the computer-related job-posting domain demonstrating that predictive rules acquired by applying KDD to an extracted database can be used to improve the recall of information extraction.

The remainder of the paper is organized as follows. Section 2 presents some background information on text mining and LIE. Section 3 describes a system called DTEX (Discovery from Text EXtraction) that combines LIE and KDD for text mining. Section 4 presents and discuss performance gains obtained in LIE by exploiting mined prediction rules. Section 5 discusses some related work, Section 6 outlines directions for future research, and Section 7 presents our conclusions.

1 II.

2 Background: Text Mining and Information Extraction

"Text mining" is used to describe the application of data mining techniques to automated discovery of useful or interesting knowledge from unstructured text [20]. Several techniques have been proposed for text mining including conceptual structure, association rule mining, episode rule mining, decision trees, and rule induction methods. In addition, Information Retrieval (IR) techniques have widely used the "bag-of-words" model [2] for tasks such as document matching, ranking, and clustering.

The related task of information extraction aims to find specific data in natural-language text. DARPA's Message Understanding Conferences (MUC) have concentrated on LIE by evaluating the performance of participating LIE systems based on blind test sets of text documents [13]. The data to be extracted is typically given by a template which specifies a list of slots to be filled with substrings taken from the document. Figure 2 shows a (shortened) document and its filled template for an information extraction task in the job-posting domain. This template includes slots that are filled by strings taken directly from the document. Several slots may have multiple fillers for the job-posting domain as in programming languages, platforms, applications, and areas.

We have developed machine learning techniques to automatically construct information extractors for job postings, such as those listed in the USENET newsgroup misc. jobs. offered [6]. By extracting information from a corpus of such textual job postings, a structured, searchable database of jobs can be automatically constructed; thus making the data in online text more easily accessible. LIE has been shown to be useful in a variety of other applications, e.g. A Frame Work for Text Mining using Learned Information Extraction System seminar announcements, restaurant guides, university web pages, apartment rental ads, and news articles on corporate acquisitions [5,9,23].

The most related system to our approach is probably DOCUMENT EXPLORER [14] which uses automatic term extraction for discovering new knowledge from texts. However, DOCUMENT EXPLORER assumes semi-structured documents such as SGML text unlike DTEX developed for general natural-language text. Similarly, automatic text categorization has been used to map web documents to pre-defined concepts for further discovery of relationships among the identified concepts [24]. One of the limitations for these approaches is that they require a substantial amount of domain knowledge.

Several rule induction methods and association rule mining algorithms have been applied to databases of corporations or product reviews automatically extracted from the web [17,16,33]; however, the interaction between LIE and rule mining has not been addressed. Recently a probabilistic framework for unifying information extraction and data mining has been proposed [25]. In this work, a graphical model using conditional probability theory is adopted for relational data, but experimental results on this approach are yet to be gathered. A

boosted text classification system based on link analysis [12] is related to our work in spirit in that it also trLIEs to improve the underlying learner by utilizing feedback from a KDD module.

3 III. Integrating Data Mining and Information Extraction

In this section, it discusses the details of our proposed text mining framework, DTEX (Discovery from Text Extraction). We consider the task of first constructing a database by applying a learned information-extraction system to a corpus of naturallanguage documents. Then, we apply standard datamining techniques to the extracted data, discovering knowledge that can be used for many tasks, including improving the accuracy of information extraction.

4 a) The DTEX System

In the proposed framework for text mining, LIE plays an important role by pre-processing a corpus of text documents in order to pass extracted items to the data mining module. In our implementations, we used two state-of-the-art systems for learning information extractors, RAPLIER (Robust Automated Production of Information Extraction Rules) [6] and BWI (Boosted Wrapper Induction) [15]. By training on a corpus of documents annotated with their filled templates, they acquire a knowledge base of extraction rules that can be tested on novel documents. RAPLIER and BWI

5 Document Title: Web Development Engineer Location: Beaverton, Oregon

This individual is responsible for design and implementation of the web-interfacing components of the Access Base server, and general back-end development duties. A successful candidate should have experience that includes: One or more of: Solaris, Linux, IBM AIX, plus Windows/NT Programming in C/C++, Java Database access and integration: Oracle, ODBC CGI and scripting: one or more of JavaScript, VBScript, Perl, PHP, ASP Exposure to the following is a plus: JDBC, Flash/Shockwave, FrontPage and/or Cold Fusion. A BSCS and 2+ years' experience (or equivalent) is required.

6 Filled Template

? title: ? "Web Development Engineer" location: ? "Beaverton, Oregon" languages: ? "C/C++", "Java", "Javascript", "VBScript", "Perl", "PHP", "ASP" platforms: ? "Solaris", "Linux", "IBM AIX", "Windows/NT" applications: ? "Oracle", "ODBC", "JDBC", "Flash/Shockwave", "FrontPage", "Cold Fusion" areas: ? "Database", "CGI", "scripting" degree required: ? "BSCS" years of experLIEnce: "2+ years" "ActiveX" "Active X" "AI" "Aritificial 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", "Micro focus Cobol"

? Oracle

Job postings (600)

? application and QA partner ?application ? SQL ?language ? HTML? language and Windows ?platform and Active Server pages ?application ? data base ? area.

? Java ?language and Active X ? area and Graphics ?area ? Web ? area ? UNIX ?platform and Windows ?platform and Games ?are ? 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 ? platform. After constructing an LIE 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 LIE 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, such as "graphics ?area", and learn rules for predicting each feature from all other features.

Similar slot fillers are first collapsed into a predetermined standard term. For example, "Windows XP" is a popular filler for the platforms slot, but it often appears as "Win XP", "Win XP", 'MS Win XP", and so on. These terms are collapsed to unique slot values before rules are mined from the data. In our experiment, a manually-constructed synonym dictionary with 111 entries was employed. Table 1 shows the first 10 entries of the dictionary. We have applied C4.5 RULES [34] to discover interesting rules from the resulting binary data. knowledge describing the relationships between slot values is written in the form of production rules. If there is a tendency for "Web" to appear in the area slot when "Director" appears in the applications slot, this is represented by the production rule, "Director .

Web". Rules can also predict the absence of a filler in a slot; however, here it focusses on rules predicting the presence of fillers. Since any LIE or KDD module can be plugged into the DTEX system, we also tested a highly-accurate information extractor (wrapper) manually developed for a book recommending system [28] to find interesting patterns from a corpus of book descriptions. Sample association rules mined from a collection of

1,500 science fiction (SF) book descriptions from the online Amazon.com bookstore are shown in Figure 5. Slots such as authors, titles, subjects, related books, and average customer ratings are identified from the corpus.

7 a) 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.

For the dataset, 600 computer-science job postings to the newsgroup austin. jobs were collected and manually annotated with correct extraction templates. Ten-fold cross validation was used to generate training and test sets. RAPLIER was used to learn the LIE component and RIPPER was used as the KDD component. Rules were induced for predicting the fillers of the languages, platforms, applications, and areas slots, since these are usually filled with multiple discrete-valued fillers and have obvious potential relationships between their values (See [30] for more details on this experiment).

In order to test the accuracy of the discovered rules, they are used to predict the information in a database of user-labelled examples. For each test document, each possible slot-value is predicted to be present or absent given information on all of its other slot-values. Average performance across all features and all test examples were then computed.

The classification accuracy for predicting the absence or presence of slot fillers is not a particularly informative performance metric since high accuracy can be ach LI Eved by simply assuming every slot filler is absent. This is because the set of potential slot fillers is very large and only a small fraction of possible fillers is present in any given example. Therefore, we evaluate the performance of DTEX using the LIE performance metrics of precision, recall, and F-measure with regard to predicting slot fillers. These metrics are defined as follows:????????????????? = ?? ?? (1)

8 Number of actual slot values correctly predicted recall= Number of actual slot values

We also report F-measure which is the harmonic mean of recall and precision:

9 F-measures= precision recall precision recall

10 × × ×

(3) Before constructing a database using an LIE system, we filtered out irrelevant documents from the newsgroup using a bag-of-words Naive-Bayes text categorizer [26]. 200 positive documents (computerscience job postings) and 20 negative examples (spam postings, resume´s, or non-cs job postings) are provided to the classifier for training. The performance of the classifier trained to predict the class" relevant" was reasonably good; precision is about 96% and recall is about 98%.

RAPLIER was trained on only 60 labelled documents, at which point its accuracy at extracting information is somewhat limited; extraction precision is about 91.9% and extraction recall is about 52.4% . We purposely trained RAPLIER on a relatively small corpus in order to demonstrate that labelling only a relatively small number of documents can result in a good set of extraction rules that is capable of building a database from which accurate knowledge can be discovered. The overall architecture of the final system is shown in Figure 6.

Figure 7 shows the learning curves for precision, recall, and F-measure of both system as well as a random guessing strategy used as a baseline. The random guessing method predicts a slot value based on its frequency of occurrence in the training data. Even with a small amount of user-labelled data, the results indicate that DTEXachieves a performance fairly comparable to the rule-miner trained on a manually constructed database.

11 IV.

12 Mined Rules to Improve Lie

After mining knowledge from extracted data, DTEX can predict information missed by the previous extraction using discovered rules. In this section, we discuss how to use mined knowledge from extracted data to aid information extraction itself. Many extraction systems provide relatively high precision, but recall is typically much lower. Previous experiments in the job postings domain showed RAPLIER's precision (e.g. low 90%'s) is significantly higher than its recall (e.g. mid 60%'s) [6]. Currently, RAPLIER's search focuses on finding high-precision rules and does not include a method for trading-off precision and recall. Although several methods have been developed for allowing a rule learner to trade-off precision and recall [11], this typically leaves the overall F-measure unchanged.

By using additional knowledge in the form of prediction rules mined from a larger set of data automatically extracted from additional unannotated text, it may be possible to improve recall without unduly sacrificing precision. For example, suppose we discover the rule "Voice XML " "Mobile". If the LIE system extracted "VoiceXML" but failed to extract "Mobile", we may want to assume there was an extraction error and add "Mobile" to the area slot, potentially improving recall. Therefore, after applying extraction rules to a document, DTEXapplies its mined rules to the resulting initial data to predict additional potential extractions.

First, we show the pseudocode for the rule mining phase in Figure 8 The extraction algorithm which attempts to improve recall by using the mined rules is summarized in Figure 9. Note that the final decision whether or not to extract a predicted filler is based on whether the filler (or any of its synonyms) occurs in the document as a substring. If the filler is found in the text, the extractor considers its prediction confirmed and extracts the filler.

One final issue is the order in which prediction rules are applLI Ed. When there are interacting rules, such as "XML Semantic Web" and "Semantic Web? areas ? . NET make the second rule fire and predict ".NET areas ? ". However, if the first rule is executed first and its prediction is confirmed, then "Semantic Web" will be extracted and the second rule can no longer fire. In DTEX, all rules with negations in their antecedent conditions are applied first. This ordering strategy attempts to maximally increase recall by making as many confirmable predictions as possible.

To summarize, documents which the user has annotated with extracted information, as well as unsupervised data which has been processed by the initial LIE system (which RAPLIER has learned from the supervised data) 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 LIE system by proposing additional slot fillers whose presence in the document are confirmed before adding them to final extraction template.

13 a) Evaluation

To test the overall system, 600 hand-labelled computer-science job postings to the newsgroup austin.jobs were collected. 10-fold cross validation was used to generate training and test sets. In addition, 4,000 unannotated documents were collected as additional optional input to the text miner. Rules were induced for predicting the fillers of the languages, platforms, applications, and areas slots, since these are usually filled with multiple discrete-valued fillers and have obvious potential relationships between their values. Details of this experiment are described in [29].

Figure 10 shows the learning curves for recall and F-measure. Unlabeled examples are not employed in these results. In order to clearly illustrate the impact of the amount of training data for both extraction and prediction rule learning, the same set of annotated data was provided to both RAPLIER and the rule miner. The results were statistically evaluated by a two-tailed, paired t-test. For each training set size, each pair of systems were compared to determine if their differences in recall and were statistically significant (

14 P <

). DTEX using prediction rules performs better than RAPLIER. As hypothesized, DTEX provides higher recall, and although it does decrease precision somewhat, overall F-measure is moderately increased. One interesting aspect is that DTEX retains a fixed recall advantage over RAPLIER as the size of the training set increases. This is probably due to the fact that the increased amount of data provided to the text miner also continues to improve the quality of the acquired prediction rules. Overall, these results demonstrate the role of data mining in improving the performance of LIE.

Table 2 shows results on precision, recall and F-measure when additional unlabeled documents are used to construct a larger database prior to mining for prediction rules. The 540 labelled 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. As a baseline, in the last row of Table 2, we also show the performance of a simple method for increasing recall by always extracting substrings that are known fillers for a particular slot. Whenever a known filler string, e.g. "C#", is contained in a test document, it is extracted as a filler for the corresponding slot, e.g. language. The reason why this works poorly is that a filler string contained in a job posting is not necessarily the correct filler for the corresponding slot. For instance, "HTML" can appear in a newsgroup posting, not in the list of required skills of that particular job announcement, but in the general instructions on submitting resume´s.

V.

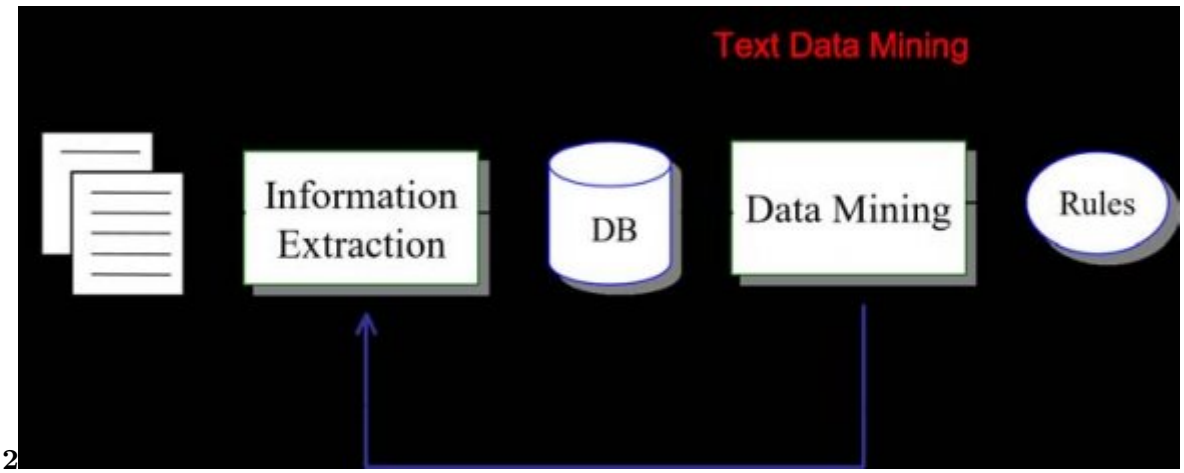
15 Conclusions

In this paper, it is presented an approach that uses an automatically learned LIE system to extract a structured database from a text corpus, and then mines this database with existing KDD tools. Our preliminary experimental results demonstrate that Learned information extraction and data mining can be integrated for the mutual benefit of both tasks. LIE enables the application of KDD to unstructured text corpora and KDD can discover predictive rules useful for improving LIE performance.

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. In particular, the growing interaction between computational linguistics and machine learning [8] is critical to the development of effective text-mining systems. [1] [2] [3]

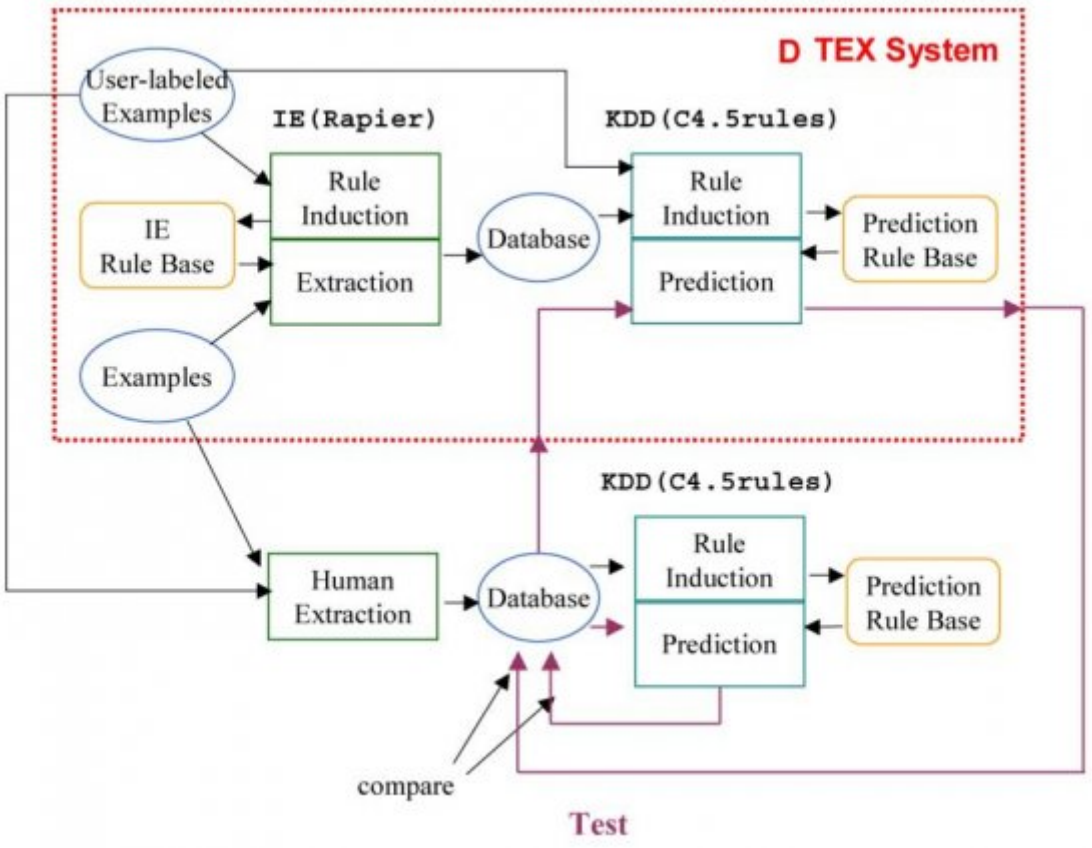
1

Figure 1: Figure 1 :



2

Figure 2: Figure 2 :



3

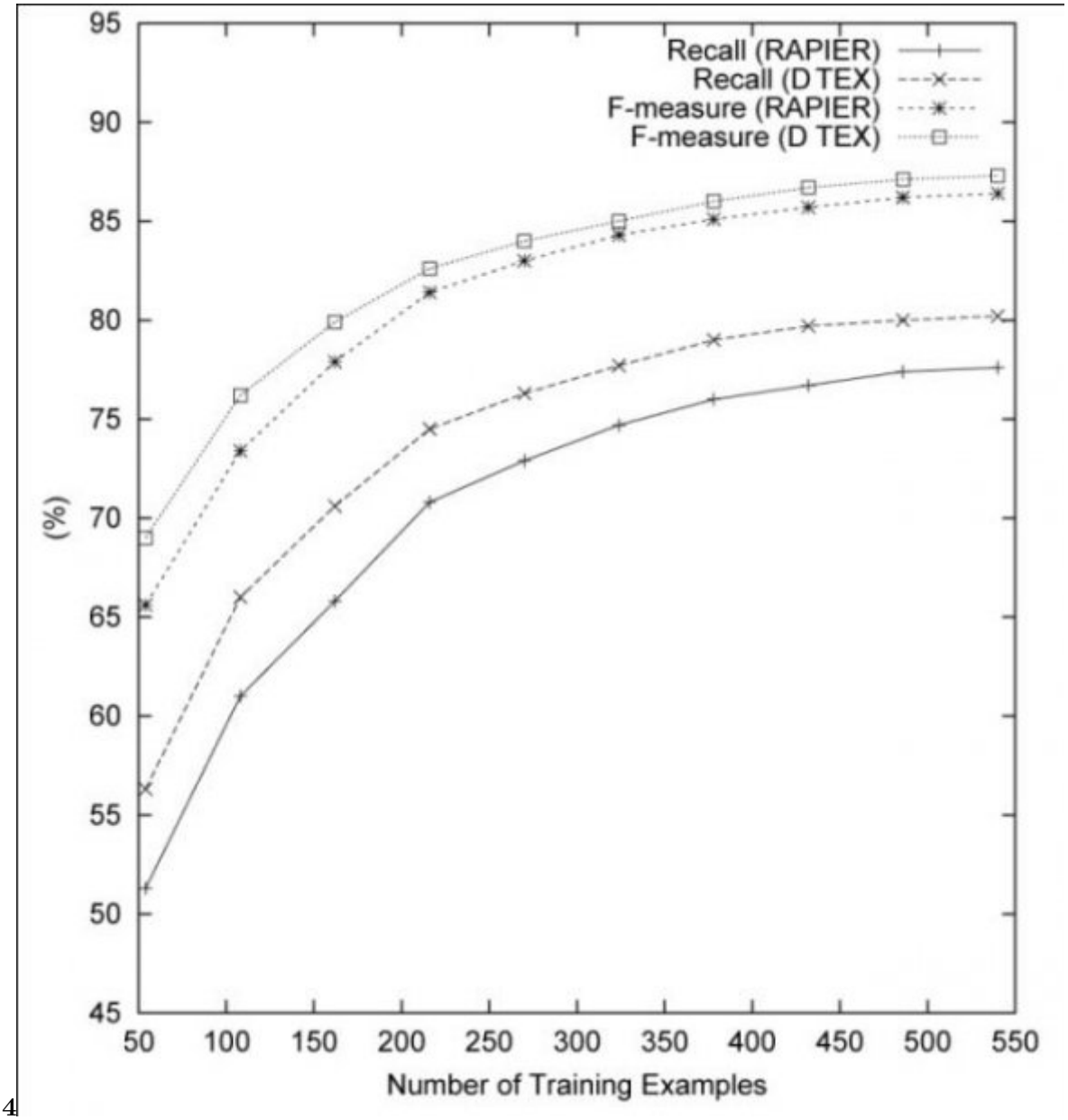
Figure 3: Figure 3 :

[1]© 2016 Global Journals Inc. (US) 1

[2]© 2016 Global Journals Inc. (US)A Frame Work for Text Mining using Learned Information Extraction System

[3]© 2016 Global Journals Inc. (US)

Figure 4: ?



4

Figure 5: Figure 4 :

1

A Frame Work for Text Mining using Learned Information Extraction System

Standard Term	Synonyms
"Access"	"MS Access", "Microsoft Access"

Year 2016
40
Volume XVI Issue III Version I
( ) C
Global Journal of Computer Science and Technology

Figure 6: Table 1 :

A Frame Work for Text Mining using Learned Information Extraction System
Year 2016
44
Volume XVI Issue III Version I
( ) C
Global Journal of Computer Science and Technology

, different rule-application orderings can produce different results. Without the first rule, a document with "XML languages ? " but without "Semantic Web area ? " in its initial filled template will

Figure 7:

Number of Examples for Rule Mining	Precision	Recall	F-Measure
0	97.4	77.6	86.4
540(Labelled)	95.8	80.2	87.3
540+1000(Unlabeled)	94.8	81.5	87.6
540+2000(Unlabeled)	94.5	81.8	87.7
540+3000(Unlabeled)	94.2	82.4	87.9
540+4000(Unlabeled)	93.5	83.3	88.1
Matching Fillers	59.4	94.9	73.1

Figure 8:

2

Year 2016
45
( )

[Note: C© 2016 Global Journals Inc. (US)]

Figure 9: Table 2 :

[Mccallum and Nigam (1998)] 'A comparison of event models for naive Bayes text classification'. K Mccallum , Nigam . *Papers from the AAAI-98 Workshop on Text Categorization*, (Madison, WI) July 1998. p. .

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

[Mccallum and Jensen (2003)] 'A note on the unification of information extraction and data mining using conditional-probability, relational models'. A Mccallum , D Jensen . *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*, (the IJCAI-2003 Workshop on Learning Statistical Models from Relational DataAcapulco, Mexico) Aug. 2003.

[Baeza-Yates and Ribeiro-Neto ()] R Baeza-Yates , B Ribeiro-Neto . *Modern Information RetrLIEval*, (New York) 1999. ACM Press.

[Freitag and Kushmerick (2000)] 'Boosted wrapper induction'. D Freitag , N Kushmerick . *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, (the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)Austin, TX) July 2000. AAAI Press / The MIT Press. p. .

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

[Ciravegna and Kushmerick (2003)] F Ciravegna , N Kushmerick . *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. 2003.

[Loh et al. (2000)] 'Concept-based knowledge discovery in texts extracted from the Web'. S Loh , L K Wives , J P M De Oliveira . *SIGKDD Explorations*, July 2000. 2 p. .

[Mooney and Roy (2000)] 'Content-based book recommending using learning for text categorization'. R J Mooney , L Roy . *Proceedings of the Fifth ACM Conference on Digital LibrarLI Es*, (the Fifth ACM Conference on Digital LibrarLI EsSan Antonio, TX) June 2000. p. .

[Ghani et al. (2000)] 'Data mining on symbolic knowledge extracted from the Web'. R Ghani , R Jones , D Mladenic´ , K Nigam , S Slattery . *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, D Mladenic´ (ed.) (the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text MiningBoston, MA) Aug. 2000. p. .

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

[Cardlie ()] 'Empirical methods in information extraction'. C Cardlie . *AI Magazine* 1997. 18 (4) p. .

[Basu et al. ()] 'Evaluating the novelty of text-mined rules using lexical knowledge'. S Basu , R J Mooney , K V Pasupuleti , J Ghosh . *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, (the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)San Francisco, CA) 2001. p. .

[Agrawal and Srikant (1994)] 'Fast algorithms for mining association rules'. R Agrawal , R Srikant . *Proceedings of the 20th International Conference on Very Large Databases (VLDB-94)*, (the 20th International Conference on Very Large Databases (VLDB-94)Santiago, Chile) Sept. 1994. p. .

[Cohen ()] 'Fast effective rule induction'. W W Cohen . *Proceedings of the Twelfth International Conference on Machine Learning (ICML-95)*, (the Twelfth International Conference on Machine Learning (ICML-95)San Francisco, CA) 1995. p. .

[Cohen ()] 'Improving a page classifLIEr with anchor extraction and link analysis'. W W Cohen . *Advances in Neural Information Processing Systems 15*, S Becker, S Thrun, K Obermayer (ed.) (Cambridge, MA) 2003. MIT Press. p. .

[Feldman et al. (1998)] 'Knowledge management: A text mining approach'. R Feldman , M Fresko , H Hirsh , Y Aumann , O Liphstat , Y Schler , M Rajman . *Proceedings of Second International Conference on Practical Aspects of Knowledge Management (PAKM-98)*, U Reimer (ed.) (Second International Conference on Practical Aspects of Knowledge Management (PAKM-98)Basel, Switzerland) Oct. 1998. 10 p. .

[Cohen ()] 'Learning to classify English text with ILP methods'. W W Cohen . *Advances in Inductive Logic Programming*, L De Raedt (ed.) (Amsterdam) 1996. IOS Press. p. .

[Cardlie and Mooney ()] 'Machine learning and natural language (Introduction to special issue on natural language learning)'. C Cardlie , R J Mooney . *Machine Learning*, 1999. 34 p. .

[Plierre (2002)] 'Mining knowledge from text collections using automatically generated metadata'. J M Plierre . *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, Lecture Notes in Computer Sicnece D Karagiannis, U Reimer (ed.) (the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)VLIEnna, Austria) Dec. 2002. Springer. 2569 p. .

[Nahm and Mooney (2002)] 'Mining soft-matching association rules'. U Y Nahm , R J Mooney . *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM2002)*, (the Eleventh International Conference on Information and Knowledge Management (CIKM2002)McLean, VA) Nov. 2002. p. .

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

[Grobelnik (ed.) ()] *Proceedings of LIEEE International Conference on Data Mining (ICDM2001) Workshop on Text Mining (TextDM'2001)*, M Grobelnik (ed.) (LIEEE International Conference on Data Mining (ICDM2001) Workshop on Text Mining (TextDM'2001)San Jose, CA) 2001.

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

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

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

[Mladenic´ (ed.) (2000)] *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, D Mladenic´ (ed.) (the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text MiningBoston, MA) Aug. 2000.

[Berry (ed.) (2003)] *Proceedings of the Third SIAM International Conference on Data Mining(SDM-2003) Workshop on Text Mining*, M W Berry (ed.) (the Third SIAM International Conference on Data Mining(SDM-2003) Workshop on Text MiningSan Francisco, CA) May 2003.

[Califf and Mooney (1999)] 'Relational learning of pattern-match rules for information extraction'. M E Califf , R J Mooney . *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, (the Sixteenth National Conference on Artificial Intelligence (AAAI-99)Orlando, FL) July 1999. p. .

[Califf (ed.) ()] *Sixteenth National Conference on Artificial Intelligence (AAAI-99) Workshop on Machine Learning for Information Extraction*, M E Califf (ed.) (Orlando, FL) 1999. AAAI Press.

[Hearst (1999)] 'Untangling text data mining'. M A Hearst . *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, (the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)College Park, MD) June 1999. p. .

[Nahm and Mooney (2000)] 'Using information extraction to aid the discovery of prediction rules from texts'. U Y Nahm , R J Mooney . *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, (the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text MiningBoston, MA) Aug. 2000. p. .

[Ghani and Fano (2002)] 'Using text mining to infer semantic attirbutes for retail data mining'. R Ghani , A E Fano . *Proceedings of the 2002 LIEEE International Conference on Data Mining (ICDM-2002)*, (the 2002 LIEEE International Conference on Data Mining (ICDM-2002)Japan) Dec. 2002. p. .

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