Mining Knowledge from Text Using Information Extraction

Raymond J. Mooney and Razvan Bunescu
Department of Computer Sciences
University of Texas at Austin
1 University Station C0500
Austin, TX 78712-0233
mooney@cs.utexas.edu, razvan@cs.utexas.edu

ABSTRACT

An important approach to text mining involves the use of natural-language information extraction. Information extraction (IE) distills structured data or knowledge from unstructured text by identifying references to named entities as well as stated relationships between such entities. IE systems can be used to directly extricate abstract knowledge from a text corpus, or to extract concrete data from a set of documents which can then be further analyzed with traditional data-mining techniques to discover more general patterns. We discuss methods and implemented systems for both of these approaches and summarize results on mining real text corpora of biomedical abstracts, job announcements, and product descriptions. We also discuss challenges that arise when employing current information extraction technology to discover knowledge in text.

1. INTRODUCTION

Most data-mining research assumes that the information to be "mined" is already in the form of a relational database. Unfortunately, for many applications, available electronic information is in the form of unstructured natural-language documents rather than structured databases. Consequently, the problem of *text mining*, i.e. discovering useful knowledge from unstructured text, is becoming an increasingly important aspect of KDD.

Much of the work in text mining does not exploit any form of natural-language processing (NLP), treating documents as an unordered "bag of words" as is typical in information retrieval. The standard a *vector space model* of text represents a document as a sparse vector that specifies a weighted frequency for each of the large number of distinct words or tokens that appear in a corpus [2]. Such a simplified representation of text has been shown to be quite effective for a number of standard tasks such as document retrieval, classification, and clustering [2; 16; 66; 60].

However, most of the knowledge that might be mined from text cannot be discovered using a simple bag-of-words representation. The entities referenced in a document and the properties and relationships asserted about and between these entities cannot be determined using a standard vector-space representation. Although full natural-language understanding is still far from the capabilities of current technology, existing methods in *information extraction* (IE) are, with reasonable accuracy, able to recognize several types of entities in text and identify some relationships that are asserted between them [14; 25; 53].

```
Production of nitric oxide ( NO ) in endothelial
cells is regulated by direct interactions of
endothelial nitric oxide synthase ( eNOS ) with
effector proteins such as Ca2+ -- calmodulin .
Here we have ...  identified a novel 34 kDa protein
, termed NOSIP ( eNOS interaction protein ) , which
avidly binds to the carboxyl terminal region of the
eNOS oxygenase domain .
```

Figure 1: Medline abstract with proteins underlined.

Therefore, IE can serve an important technology for text mining. If the knowledge to be discovered is expressed directly in the documents to be mined, then IE alone can serve as an effective approach to text mining. However, if the documents contain concrete data in unstructured form rather than abstract knowledge, it may be useful to first use IE to transform the unstructured data in the document corpus into a structured database, and then use traditional data-mining tools to identify abstract patterns in this extracted data.

In this article, we review these two approaches to text mining with information extraction, using one of our own research projects to illustrate each approach. First, we introduce the basics of information extraction. Next, we discuss using IE to directly extract knowledge from text. Finally, we discuss discovering knowledge by mining data that is first extracted from unstructured or semi-structured text.

2. INFORMATION EXTRACTION

2.1 IE Problems

Information Extraction (IE) concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text. One type of IE, *named entity recognition*, involves identifying references to particular kinds of objects such as names of people, companies, and locations [4]. In this paper, we consider the task of identifying names of human proteins in abstracts of biomedical journal articles [10]. Figure 1 shows part of a sample abstract in which the protein names are underlined.

In addition to recognizing entities, an important problem is extracting specific types of relations between entities. For example, in newspaper text, one can identify that an organization is located in a particular city or that a person is

Sample Job Posting:

Job Title: Senior DBMS Consultant
Location: Dallas,TX
Responsibilities:
DBMS Applications consultant works with project teams to define DBMS based solutions that support the enterprise deployment of Electronic Commerce, Sales Force Automation, and Customer Service applications.
Desired Requirements:
3-5 years exp. developing Oracle or SQL Server apps using Visual Basic, C/C++, Powerbuilder, Progress, or similar.
Recent experience related to installing and configuring Oracle or SQL Server in both dev. and deployment environments.
Desired Skills:
Understanding of UNIX or NT, scripting language. Know principles of structured software engineering and project management

Filled Job Template:

title: Senior DBMS Consultant
state: TX
city: Dallas
country: US
language: Powerbuilder, Progress, C, C++, Visual Basic
platform: UNIX, NT
application: SQL Server, Oracle
area: Electronic Commerce, Customer Service
required years of experience: 3
desired years of experience: 5

Figure 2: Sample Job Posting and Filled Template

affiliated with a specific organization [73; 24]. In biomedical text, one can identify that a protein interacts with another protein or that a protein is located in a particular part of the cell [10; 23]. For example, identifying protein interactions in the abstract excerpt in Figure 1 would require extracting the relation: `interacts`(**NOSIP**, **eNOS**).

IE can also be used to extract fillers for a predetermined set of slots (roles) in a particular template (frame) relevant to the domain. In this paper, we consider the task of extracting a database from postings to the USENET newsgroup, `austin.jobs` [12]. Figure 2 shows a sample message from the newsgroup and the filled computer-science job template where several slots may have multiple fillers. For example, slots such as languages, platforms, applications, and areas usually have more than one filler, while slots related to the job's title or location usually have only one filler. Similar applications include extracting relevant sets of predefined slots from university colloquium announcements [29] or apartment rental ads [67].

Another application of IE is extracting structured data from unstructured or semi-structured web pages. When applied to semi-structured HTML, typically generated from an underlying database by a program on a web server, an IE system is typically called a *wrapper* [37], and the process is sometimes referred to as *screen scraping*. A typical application is extracting data on commercial items from web stores for a comparison shopping agent (*shopbot*) [27] such as MySimon (`www.mysimon.com`) or Froogle (`froogle.google.com`). For example, a wrapper may extract the title, author, ISBN number, publisher, and price of book from an Amazon web page.

IE systems can also be used to extract data or knowledge from less-structured web sites by using both the HTML text in their pages as well as the structure of the hyperlinks between their pages. For example, the WebKB project at Carnegie Mellon University has explored extracting structured information from university computer-science departments [22]. The overall WebKB system attempted to identify all faculty, students, courses, and research projects in a department as well as relations between these entities such as: instructor(prof, course), advisor(student, prof), and member(person, project).

2.2 IE Methods

There are a variety of approaches to constructing IE systems. One approach is to manually develop information-extraction rules by encoding patterns (e.g. regular expressions) that reliably identify the desired entities or relations. For example, the Suiseki system [8] extracts information on interacting proteins from biomedical text using manually developed patterns.

However, due to the variety of forms and contexts in which the desired information can appear, manually developing patterns is very difficult and tedious and rarely results in robust systems. Consequently, supervised machine-learning methods trained on human annotated corpora has become the most successful approach to developing robust IE systems [14]. A variety of learning methods have been applied to IE.

One approach is to automatically learn pattern-based extraction rules for identifying each type of entity or relation. For example, our previously developed system, RAPIER [12; 13], learns extraction rules consisting of three parts: 1) a pre-filler pattern that matches the text immediately preceding the phrase to be extracted, 2) a filler pattern that matches the phrase to be extracted, and 3) a post-filler pattern that matches the text immediately following the filler. Patterns are expressed in an enhanced regular-expression language, similar to that used in Perl [72]; and a bottom-up relational rule learner is used to induce rules from a corpus of labeled training examples. In Wrapper Induction [37] and Boosted Wrapper Induction (BWI) [30], regular-expression-type patterns are learned for identifying the beginning and ending of extracted phrases. Inductive Logic Programming (ILP) [45] has also been used to learn logical rules for identifying phrases to be extracted from a document [29].

An alternative general approach to IE is to treat it as a *sequence labeling* task in which each word (token) in the document is assigned a label (tag) from a fixed set of alternatives. For example, for each slot, X, to be extracted, we include a token label $BeginX$ to mark the beginning of a filler for X and $InsideX$ to mark other tokens in a filler for X. Finally, we include the label $Other$ for tokens that are not included in the filler of any slot. Given a sequence labeled with these tags, it is easy to extract the desired fillers.

One approach to the resulting sequence labeling problem is to use a statistical sequence model such as a Hidden Markov Model (HMM) [57] or a Conditional Random Field (CFR) [38]. Several earlier IE systems used generative HMM models [4; 31]; however, discriminately-trained CRF models have recently been shown to have an advantage over HMM's [54; 65]. In both cases, the model parameters are learned from

Pre-filler Pattern:	Filler Pattern:	Post-filler Pattern:
1) syntactic: {nn,nnp}	1) word: undisclosed	1) semantic: price
2) list: length 2	syntactic: jj	

Figure 3: Sample Extraction Rule Learned by RAPIER

a supervised training corpus and then an efficient dynamic programming method based on the Viterbi algorithm [71] is used to determine the most probable tagging of a complete test document.

Another approach to the sequence labeling problem for IE is to use a standard feature-based inductive classifier to predict the label of each token based on both the token itself and its surrounding context. Typically, the context is represented by a set of features that include the one or two tokens on either side of the target token as well as the labels of the one or two preceding tokens (which will already have been classified when labeling a sequence from left to right). Using this general approach, IE systems have been developed that use many different trained classifiers such as decision trees [3], boosting [15], memory-based learning (MBL) [43], support-vector machines (SVMs) [40], maximum entropy (MaxEnt) [17], transformation-based learning (TBL)[68] and many others [64].

Many IE systems simply treat text as a sequence of uninterpreted tokens; however, many others use a variety of other NLP tools or knowledge bases. For example, a number of systems preprocess the text with a part-of-speech (POS) tagger (e.g. [18; 9]) and use words' POS (e.g. noun, verb, adjective) as an extra feature that can be used in hand-written patterns [8], learned extraction rules [13], or induced classifiers [64]. Several IE systems use phrase chunkers (e.g. [59]) to identify potential phrases to extract [64; 73]. Others use complete syntactic parsers (e.g. [21]), particularly those which try to extract relations between entities by examining the synactic relationship between the phrases describing the relevant entities [24; 61]. Some use lexical semantic databases, such as WordNet [28], which provide word classes that can be used to define more general extraction patterns [13].

As a sample extraction pattern, Figure 3 shows a rule learned by RAPIER [13] for extracting the transaction amount from a newswire concerning a corporate acquisition. This rule extracts the value "undisclosed" from phrases such as "sold to the bank for an undisclosed amount" or "paid Honeywell an undisclosed price". The pre-filler pattern matches a noun or proper noun (indicated by the POS tags 'nn' and 'pn', respectively) followed by at most two other unconstrained words. The filler pattern matches the word "undisclosed" only when its POS tag is "adjective." The post-filler pattern matches any word in WordNet's semantic class named "price".

3. EXTRACTING KNOWLEDGE

If the information extracted from a corpus of documents represents abstract knowledge rather than concrete data, IE itself can be considered a form of "discovering" knowledge from text. For example, an incredible wealth of biological knowledge is stored in published articles in scientific journals. Summaries of more than 11 million such articles are available in the Medline database;[1] however, retrieving and processing this knowledge is very difficult due to the lack of formal structure in the natural-language narrative in these documents. Automatically extracting information from biomedical text holds the promise of easily consolidating large amounts of biological knowledge in computer-accessible form. IE systems could potentially gather information on global gene relationships, gene functions, protein interactions, gene-disease relationships, and other important information on biological processes. Consequently, a growing number of recent projects have focused on developing IE systems for biomedical literature [33; 23; 7; 55; 32].

We recently compared many of the IE methods introduced in section 2.2 on the task of extracting human protein names [10; 58]. Specifically, we evaluated RAPIER, BWI, MBL, TBL, SVM, MaxEnt, HMM, and CRF by running 10-fold cross-validation on set of approximately 1,000 manually-annotated Medline abstracts that discuss human proteins. Overall, we found that CRF's gave us the best result on this particular problem. However, although CRF's capture the dependence between the labels of adjacent words, it does not adequately capture long-distance dependencies between potential extractions in different parts of a document. For example, in our protein-tagging task, repeated references to the same protein are common. If the context surrounding one occurrence of a phrase is very indicative of it being a protein, then this should also influence the tagging of another occurrence of the same phrase in a different context which is not typical of protein references. Therefore, we recently developed a new IE method based on Relational Markov Networks (RMN's) [69] that captures dependencies between distinct candidate extractions in a document [11]. Experimental evaluation confirmed that this approach increases accuracy of human-protein recognition compared to a traditional CRF.

We have also evaluated several approaches to extracting protein interactions from text in which protein names have already been identified [10]. Blaschke *et al.* [7; 8] manually developed patterns for extracting interacting proteins, where a pattern is a sequence of words (or POS tags) and two protein-name tokens. Between every two adjacent words is a number indicating the maximum number of arbitrary words that can be skipped at this position. Therefore, we developed a new IE learning method, ELCS [10], that automatically induces such patterns using a bottom-up rule learning algorithm that computes generalizations based on *longest common subsequences* [35]. Below is a sample pattern that it learned for extracting protein interactions:

```
- (7) interactions (0) between (5) PROT (9)
   PROT (17) .
```

where `PROT` matches a previously tagged protein name. The induced patterns were able to identify interactions more precisely than the human-written ones.

Another approach we have taken to identifying protein interactions is based on *co-citation* [39]. This approach does not try to find specific assertions of interaction in the text, but rather exploits the idea that if many different abstracts reference both protein A and protein B, then A and B are likely to interact. Particularly, if two proteins are co-cited significantly more often than one would expect if they were cited independently at random, then it is likely that they interact. In order to account for the case where the co-citing abstracts

[1] Available on the web at `http://www.ncbi.nlm.nih.gov/entrez/`

do not actually concern protein interactions but cite multiple proteins for other reasons, we also used a Bayesian "bag of words" text classifier trained to discriminate between abstracts that discuss protein interactions from those that do not [39]. In order to find interactions, protein pairs that are highly co-cited were filtered for those which are specifically co-cited in abstracts that the Bayesian text-classifier assigns high-probability of discussing protein interactions.

Using these techniques, we recently completed the initial phase of a large-scale project to mine a comprehensive set of human protein interactions from the biomedical literature. By mining 753,459 human-related abstracts from Medline with a combination of a CRF-based protein tagger, co-citation analysis, and automatic text classification, we extracted a set of 6,580 interactions between 3,737 proteins. By utilizing information in existing protein databases, this automatically extracted data was found to have an accuracy comparable to manually developed data sets. Based on comparisons to these existing protein databases, the co-citation plus text-classification approach was found to be more effective at identifying interactions than our IE approach based on ELCS. We are currently developing an improved IE system for identifying direct assertions of interactions in biomedical text using an SVM that exploits a relational string kernel similar to that utilized by Zelenko *et al.* [73].

By consolidating our text-mined knowledge with existing manually-constructed biological databases, we have assembled a large, fairly comprehensive, database of known human protein interactions containing 31,609 interactions amongst 7,748 proteins. More details on our database of protein interactions have been published in the biological literature [58] and it is freely available on the web.[2] Therefore, using automated text mining has helped build an important knowledge base of human proteins that has been recognized as a contribution worthy of publication in *Genome Biology* and will hopefully become a valuable resource to biologists.

4. MINING EXTRACTED DATA

If extracted information is specific data rather than abstract knowledge, an alternative approach to text mining is to first use IE to obtain structured data from unstructured text and then use traditional KDD tools to discover knowledge from this extracted data. Using this approach, we developed a text-mining system called DiscoTEX (Discovery from Text EXtraction) [48; 49] which has been applied to mine job postings and resumés posted to USENET newsgroups as well as Amazon book-description pages spidered from the web.

In DiscoTEX, IE plays the important role of preprocessing a corpus of text documents into a structured database suitable for mining. DiscoTEX uses two learning systems to build extractors, Rapier [12] and BWI [30]. By training on a corpus of documents annotated with their filled templates, these systems acquire pattern-matching rules that can be used to extract data from novel documents.

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 extractor 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, DiscoTEX induces 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 $\in$ *area*", and learn rules for predicting each feature from all other features.

We have applied C4.5rules [56] to discover interesting rules from the resulting binary data. Discovered 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 $\in$ *application* $\rightarrow$ Web $\in$ *area*". Sample rules that C4.5rules mined from a database of 600 jobs that Rapier extracted from the USENET newsgroup `austin.jobs` are shown in Figure 4. The last rule illustrates the discovery of an interesting concept which could be called "the IBM shop;" i.e. companies that require knowledge of an IBM operating system and DBMS, but not a competing DBMS, also require knowledge of Lotus Notes, another IBM product.

We also applied Ripper [20] and Apriori [1] to discover interesting rules from extracted data. Sample rules mined from a database of 600 resumés extracted from the USENET newsgroup `misc.jobs.resumes` by BWI are shown in Figure 5. The first two rules were discovered by Ripper while the last two were found by Apriori. Since any IE or KDD module can be plugged into the DiscoTEX system, we also used an information extractor (wrapper) manually developed for a book recommending system [44] to find interesting patterns in a corpus of book descriptions. Sample association rules mined from a collection of 1,500 science fiction book descriptions from the online `Amazon.com` bookstore are shown in Figure 6.

Ghani *et al.* [34], have developed a system similar to DiscoTEX and used it to mine data about companies extracted from the web. One of the patterns they discovered was "Advertising companies are usually located in New York."

Unfortunately, the accuracy of current IE systems is limited,

- Oracle $\in$ *application* and QA Partner $\in$ *application* $\rightarrow$ SQL $\in$ *language*
- Java $\in$ *language* and ActiveX $\in$ *area* and Graphics $\in$ *area* $\rightarrow$ Web $\in$ *area*
- $\neg$(UNIX $\in$ *platform*) and $\neg$(Windows $\in$ *platform*) and Games $\in$ *area* $\rightarrow$ 3D $\in$ *area*
- AIX $\in$ *platform* and $\neg$(Sybase $\in$ *application*) and DB2 $\in$ *application* $\rightarrow$ Lotus Notes $\in$ *application*

Figure 4: Sample rules mined from CS job postings.

- HTML $\in$ *language* **and** DHTML $\in$ *language* $\rightarrow$ XML $\in$ *language*
- Dreamweaver 4 $\in$ *application* **and** Web Design $\in$ *area* $\rightarrow$ Photoshop 6 $\in$ *application*
- ODBC $\in$ *application* $\rightarrow$ JSP $\in$ *language*
- Perl $\in$ *language* **and** HTML $\in$ *language* $\rightarrow$ Linux $\in$ *platform*

Figure 5: Sample rules mined from CS resumés.

[2] `http://bioinformatics.icmb.utexas.edu/idserve`

- Sign of the Unicorn $\in$ *related books* **and** American Science Fiction $\in$ *subject* $\Rightarrow$ Knight of Shadows $\in$ *related books*
- Spider Robinson $\in$ *author* $\Rightarrow$ Jeanne Robinson $\in$ *author*
- Roger Zelazny $\in$ *author* $\Rightarrow$ 5 $\in$ *average rating*

Figure 6: Sample rules mined from book descriptions.

and therefore an automatically extracted database will inevitably contain a fair number of errors. An important question is whether the knowledge discovered from a "noisy" extracted database is significantly less reliable than knowledge discovered from a "clean" manually-constructed database. We have conducted experiments on job postings showing that rules discovered from an automatically extracted database are very close in accuracy to those discovered from a corresponding manually-constructed database [49]. These results demonstrate that mining extracted data is a reliable approach to discovering accurate knowledge from unstructured text.

Another potential problem with mining extracted data is that the heterogeneity of extracted text frequently prevents traditional data-mining algorithms from discovering useful knowledge. The strings extracted to fill specific data fields can vary substantially across documents even though they refer to the same real-world entity. For example, the Microsoft operating system may be referred to as "Windows," "Microsoft Windows," "MS Windows," etc.. We developed two approaches to addressing this problem [47].

One approach is to first "clean" the data by identifying all of the extracted strings that refer to the same entity and then replacing sets of equivalent strings with canonical entity names. Traditional KDD tools can then be used to mine the resulting "clean" data. Identifying textually distinct items that refer to the same entity is an instance of the general database "deduping" or record-linkage problem [42; 5]. We have developed methods for learning string-similarity measures and classifiers that identify equivalent items by training on a sample of manually "deduped" data [6].

Another approach to handling heterogeneity is to mine "soft matching" rules directly from the "dirty" data extracted from text. In this approach, the rule induction process is generalized to allow partial matching of data strings in order to discover important regularities in variable textual data. We developed two novel methods for mining soft-matching rules. First, is an algorithm called TextRISE [3] that learns rules whose conditions are partially matched to data using a similarity metric [50]. We also developed SoftApriori, a generalization of the standard Apriori algorithm for discovering association rules [1] that allows soft matching using a specified similarity metric for each field [51]. In both methods, similarity of textual items is measured using either standard "bag of words" metrics [63] or edit-distance measures [35]; other standard similarity metrics can be used for numerical and other data types. Experimental results in several domains have demonstrated that both TextRISE and SoftApriori allow the discovery of interesting "soft-matching" rules from automatically-extracted data. These rules accurately capture regularities not discovered by traditional methods [46].

[3]A variant of Domingos' RISE algorithm [26], an elegant approach to unifying rule-based and instance-based methods.

5. FUTURE RESEARCH

Information extraction remains a challenging problem with many potential avenues for progress. In section 4, we discussed mining knowledge from extracted data; this discovered knowledge can itself be used to help improve extraction. 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 discover the rule "MySQL$\in$ *language*" $\rightarrow$ "Database $\in$ *area*". If the IE system extracted "MySQL $\in$ *language*" but failed to extract "Database $\in$ *area*", we may want to assume there was an extraction error and add "Database" to the area slot. We have developed methods for using mined knowledge to improve the recall of extraction but not the precision [48; 52]. McCallum and Jensen [41] propose using probabilistic graphical models to unify IE and KDD; however, actual results on this approach are a goal of on-going research.

Most IE systems are developed by training on human annotated corpora; however, constructing corpora sufficient for training accurate IE systems is a burdensome chore. One approach is to use active learning methods to decrease the amount of training data that must be annotated by selecting only the most informative sentences or passages to give to human annotators. We presented an initial approach to active learning for IE [70]; however, more research is needed to explore methods for reducing the demand for supervised training data in IE.

Another approach to reducing demanding corpus-building requirements is to develop unsupervised learning methods for building IE systems. Although some work has been done in this area [19; 36], this is another promising area for future research. Developing semi-supervised learning methods for IE is a related research direction in which there has been only a limited amount of work [62].

With respect to handling textual variation when mining extracted data, it would be nice to see experimental comparisons of the two approaches mentioned in section 4; i.e. automated data cleaning versus mining "soft matching" rules from "dirty" data. Do both approaches discover equally accurate knowledge with similar computational efficiency?

When mining "soft-matching" rules, our current methods use a fixed, predetermined similarity metric for matching rule antecedents to variable text data. However, we have developed adaptive learned similarity metrics for data cleaning and "deduping' [6]. It would be interesting to use such learned similarity metrics when discovering "soft-matching" rules since judging the similarity of textual strings is often domain dependent.

6. CONCLUSIONS

In this paper we have discussed two approaches to using natural-language information extraction for text mining. First, one can extract general knowledge directly from text. As an example of this approach, we reviewed our project which extracted a knowledge base of 6,580 human protein interactions by mining over 750,000 Medline abstracts. Second, one can first extract structured data from text documents or web pages and then apply traditional KDD methods to discover patterns in the extracted data. As an example of this approach, we reviewed our work on the DiscoTEX system and its application to Amazon book descriptions and

computer-science job postings and resumés.
Research in information extraction continues to develop more effective algorithms for identifying entities and relations in text. By exploiting the lastest techniques in human-language technology and computational linguistics and combining them with the latest methods in machine learning and traditional data mining, one can effectively mine useful and important knowledge from the continually growing body of electronic documents and web pages.

7. ACKNOWLEDGEMENTS

Our research on bioinformatics extraction was done in collaboration with Arun Ramani, Edward Marcotte, Ruifang Ge, Rohit Kate, and Yuk Wah Wong. The DISCOTEX system was developed primarily by Un Yong Nahm. This research was partially supported by the National Science Foundation under grants IIS-0325116 and IIS-0117308.

8. 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. W. Bennett, C. Aone, and C. Lovell. Learning to tag multilingual texts through observation. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, pages 109–116, Providence, RI, 1997.

[4] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34:211–232, 1999.

[5] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[6] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 39–48, Washington, DC, Aug. 2003.

[7] C. Blaschke and A. Valencia. Can bibliographic pointers for known biological data be found automatically? protein interactions as a case study. *Comparative and Functional Genomics*, 2:196–206, 2001.

[8] C. Blaschke and A. Valencia. The frame-based module of the Suiseki information extraction system. *IEEE Intelligent Systems*, 17:14–20, 2002.

[9] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[10] R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine (special issue on Summarization and Information Extraction from Medical Documents)*, 33(2):139–155, 2005.

[11] R. C. Bunescu and R. J. Mooney. Collective information extraction with relational Markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 439–446, Barcelona, Spain, July 2004.

[12] 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 1999.

[13] M. E. Califf and R. J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, 4:177–210, 2003.

[14] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–79, 1997.

[15] X. Carreras, L. Màrquez, and L. Padró. A simple named entity extractor using AdaBoost. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, Edmonton, Canada, 2003.

[16] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, CA, 2002.

[17] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 160–163, Edmonton, Canada, 2003.

[18] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, TX, 1988. Association for Computational Linguistics.

[19] F. Ciravegna, A. Dingli, D. Guthrie, and Y. Wilks. Mining web sites using unsupervised adaptive information extraction. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, Apr. 2003.

[20] 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, 1995.

[21] M. J. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 16–23, 1997.

[22] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 509–516, Madison, WI, July 1998.

[23] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-1999)*, pages 77–86, Heidelberg, Germany, 1999.

[24] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, July 2004.

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

[26] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.

[27] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In *Proceedings of the First International Conference on Autonomous Agents (Agents-97)*, pages 39–48, Marina del Rey, CA, Feb. 1997.

[28] C. D. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[29] D. Freitag. Toward general-purpose learning for information extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and COLING-98 (ACL/COLING-98)*, pages 404–408, Montreal, Quebec, 1998.

[30] 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 2000. AAAI Press / The MIT Press.

[31] D. Freitag and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, 2000. AAAI Press / The MIT Press.

[32] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17:S74–S82, 2001. Supplement 1.

[33] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Information extraction: Identifying protein names from biological papers. In *Proceedings of the 3rd Pacific Symposium on Biocomputing*, pages 707–718, 1998.

[34] 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.

[35] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, New York, 1997.

[36] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 416–423, Barcelona, Spain, July 2004.

[37] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 729–735, Nagoya, Japan, 1997.

[38] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning (ICML-2001)*, pages 282–289, Williamstown, MA, 2001.

[39] E. Marcotte, I. Xenarios, and D. Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, Apr;17(4):359–363, 2001.

[40] J. Mayfield, P. McNamee, and C. Piatko. Named entity recognition using hundreds of thousands of features. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, Edmonton, Canada, 2003.

[41] 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. 2003.

[42] A. McCallum, S. Tejada, and D. Quass, editors. *Proceedings of the KDD-03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, Washington, DC, Aug. 2003.

[43] F. D. Meulder and W. Daelemans. Memory-based named entity recognition using unannotated data. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, Edmonton, Canada, 2003.

[44] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 195–204, San Antonio, TX, June 2000.

[45] S. H. Muggleton, editor. *Inductive Logic Programming*. Academic Press, New York, NY, 1992.

[46] U. Y. Nahm. *Text Mining with Information Extraction*. PhD thesis, Department of Computer Sciences, University of Texas, Austin, TX, Aug. 2004.

[47] U. Y. Nahm, M. Bilenko, and R. J. Mooney. Two approaches to handling noisy variation in text mining. In *Papers from the Nineteenth International Conference on Machine Learning (ICML-2002) Workshop on Text Learning*, pages 18–27, Sydney, Australia, July 2002.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] U. Y. Nahm and R. J. Mooney. Using soft-matching mined rules to improve information extraction. In *Proceedings of the AAAI-2004 Workshop on Adaptive Text Extraction and Mining (ATEM-2004)*, pages 27–32, San Jose, CA, July 2004.

[53] National Institute of Standards and Technology. ACE - Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/.

[54] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting (HLT-NAACL-2004)*, Boston, MA, 2004.

[55] C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Association of genes to genetically inherited diseases using data mining. *Nature Genetics*, 31(3):316–319, July 2002.

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

[57] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[58] A. K. Ramani, R. C. Bunescu, R. J. Mooney, and E. M. Marcotte. Consolidating the set of know human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biology*, 6(5):r40, 2005.

[59] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, 1995.

[60] E. M. Rasmussen. Clustering algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[61] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pages 1273–1279, Seattle, WA, 2001.

[62] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, OR, 1996.

[63] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[64] E. F. T. K. Sang and F. D. Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, Edmonton, Canada, 2003.

[65] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17*, Vancouver, Canada, 2005.

[66] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[67] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34:233–272, 1999.

[68] L. Tanabe and W. J. Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.

[69] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of 18th Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 485–492, Edmonton, Canada, 2002.

[70] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pages 406–414, Bled, Slovenia, June 1999.

[71] A. J. Viterbi. Error bounds for convolutional codes and and asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[72] L. Wall, T. Christiansen, and R. L. Schwartz. *Programming Perl*. O'Reilly and Associates, Sebastopol, CA, 1996.

[73] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.