Knowl Inf Syst (2011) 29:159–201
DOI 10.1007/s10115-011-0439-8

REGULAR PAPER

Finding best evidence for evidence-based best practice recommendations in health care: the initial decision support system design

Nick Cercone · Xiangdong An · Jiye Li · Zhenmei Gu · Aijun An

Received: 28 July 2008 / Revised: 13 June 2011 / Accepted: 8 August 2011 /
Published online: 24 August 2011

Abstract A major problem for Canadian health organizations is finding best evidence for evidence-based best practice recommendations. Medications are not always effectively used and misuse may harm patients. Drugs are the fastest-growing element of Canadian health care spending, second only to hospital spending. Three hundred million prescriptions are filled annually. Prescription drugs accounted for 5.8% of total health care spending in 1980 and close to 18% today. A primary long-term goal of this research is to develop a decision support system for evidence-based management, quality control and best practice recommendations for medical prescriptions. Our results will improve accessibility and management of information by: (1) building an prototype for adaptive information extraction, text and data mining from (online) documents to find evidence on which to base best practices; and (2) employing multiply sectioned Bayesian networks (MSBNs) to infer a probabilistic interpretation to validate evidence for recommendations; MSBNs provide this structure. Best practices to improve drug-related health outcomes; patients' quality of life; and cost-effective use of medications by changing knowledge and behavior. This research will support next generation eHealth decision support systems, which routinely find and verify evidence from multiple sources, leading to cost-effective use of drugs, improve patients' quality of life and optimize drug-related health outcomes.

Keywords Evidence-based best practices · Decision support · Naive Bayes · Segment-based hidden Markov models (HMMs) · Multiply sectioned Bayes nets · Association rules · Reduct · Information extraction · Data mining

1 Introduction

Medications are not always effectively used and misuse may harm patients. Drugs are the fastest-growing element of Canadian health care spending, second only to hospital spending.

N. Cercone (✉) · X. An · J. Li · Z. Gu · A. An
Department of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada
e-mail: ncercone@yorku.ca

Prescription drugs play an increasingly important role in Canada's health system. 300 million prescriptions are filled each year or an average of 10 per person per year. Prescription drugs accounted for 5.8% of total health care spending in 1980 and 13% of total health care spending in 2003.[1] The rising cost of prescription drugs is clearly the most unsustainable part of Canadian Provincial health care spending and needs to be controlled.[2]

A primary long-term goal of this research is to develop a decision support system for evidence-based management, quality control and best practice recommendations for medical prescriptions [41]. Our research will improve information access, management and handling in a networked environment (cf. [48]) by:

1. building an application prototype for adaptive information extraction (IE), text and data mining from (online) documents to find evidence on which to base best practice recommendations; and
2. employing multiply sectioned Bayesian networks (MSBNs) to infer a probabilistic interpretation to validate evidence for recommendations (MSBNs provide a framework for probabilistic inference).

We collaborate with the Canadian optimal medication prescribing and utilization service (COMPUS), a nationally coordinated program that defines best practices as an approach to drug prescribing or use that, based on the evidence, is clinically and cost-effective; and contributes to optimal health outcomes. Best practice activities are aimed at improving drug-related health outcomes; patients' quality of life; and the cost-effective use of medications by changing knowledge, attitudes and behavior. Evidence is based on research from published and unpublished literature (from systematic reviews or meta-analyses of randomized control trials, controlled observational studies and, case series and expert opinions).

Decisions on recommendations for clinical practice are based upon different information sources. The question that naturally arises is this: how do we tell "good evidence" from "bad evidence"[3] and how much evidence is sufficient to make a recommendation?

COMPUS systematically gathers recommendations and evidence supporting those recommendations from different clinical guideline documents according to a well-established methodology. Most of this evidence was generated under controlled ideal conditions. Such recommendations and related evidence is then presented to a group of 11–12 experts who review the evidence and select the best recommendations as best practices based upon the evidence and their clinical experience.

The overall time frame for gathering the evidence and checking the internal and external validity is approximately 3 months and is very labor intensive. The expert panel typically spends 1–2 h in discussion subjectively evaluating the evidence before making recommendations.

In this phase of research, we conduct three investigations, the combination of which will support achievement of the two parts of our primary long-term goal presented above.

[1] https://www.ccohta.ca/compus/compus_presentations/compus_ConsultationSessions_Presentation_Barb%20Wells.pdf.

[2] http://cupe.ca/Drugs/How_rising_drug_cost.

[3] Evidence could be considered a rule or a group of related events extracted from text. Evidence might be "good" if the same rule can be extracted from many documents since it is supported widely. A rule might also be considered "good" if it is consistent with the experts' opinions. The former can be evaluated based on the amount of supporting documents for the rule by IE. The latter could be evaluated using MSBNs. With MSBNs, we model the knowledge of a group of medical experts. We then test the rule on the MSBN to see whether the rule is well interpretable. If very contradictory or inconsistent results could be derived from the model, then the rule might not be "good".

These three investigations are described in Sects. 3, 4, 5, 6 and portray our approach, research methodology and impact this research will have. The decision support system for evidence-based management has not yet been built as we continue to develop and integrate the techniques described in Sects. 3, 4, 5, 6.

Examples of finding evidence by IE can be found in Sects. 4 and 5. Section 4 considers segment-based hidden Markov models for IE; Section 5 illustrates how to generate important rules directly from the database where each record can be considered raw evidence obtained by IE.

For example, all sample rules in Table 4 can be considered evidence obtained by machine-learning and IE techniques from a Geriatric Care Data Set supplied by Dalhousie University Medical Faculty. Database details are presented in Sect. 5, Table 3. Important rules are extracted from the database based on a subset of attributes representative of the entire database (called a reduct) without loss of generality and we explain clearly in Sect. 5 different tasks, e.g., finding important rules from medical data.

Another example can be found in [62] which answers the question "For a 54-year-old woman with periodontal disease, how effective is the therapeutic use of doxcyline to decrease gum bleeding and recession compared to no treatment?"

2 Research preliminaries, techniques and methods

Healthcare's most valuable information (evidence) is found trapped inside the free-form texts of patient histories and progress notes. Much of medicine involves language data and the bulk of medical data is narrative [27]. Physicians want to use both structured input and narrative text—which implies text processing and natural language analysis [2].

Controlling medical terminologies using standards for classifying and coding medical information such as the systematized nomenclature of medicine (SNOMED) cannot analyze everything in healthcare's free-form documentation. A first step will be to build a system to extract elements from narrative text for comparison with other data, and can provide clinicians' verification of evidence found in evidence-based best practice recommendations. Such clinical decision support may provide clinicians/patients information about incomplete documentation or catch medication errors by providing feedback to the physicians.

2.1 Facts, factoids, frames

We focus on improving the robustness of existing IE systems (e.g., WHISK [55], RAPIER [14], SRV [17] and LP2 [16]) which automatically learn extraction knowledge from data, and ease the difficulty of adapting an IE system to different extraction tasks [21]. We examine the naive Bayes IE model as a purely adaptive IE model, in which the "formulation problem" existing in previous naive Bayes IE systems is corrected. We investigate the effect of smoothing techniques in this context (essentially a general issue associated with any probabilistic model), and design our own smoothing strategy [23] to obtain more stable probability estimation in statistical IE learning. Our initial experimental results show that a good smoothing [22] method is critical to the robustness of naive Bayes IE systems.

In most existing probabilistic systems, a natural evolution from the naive Bayes' models is to more advanced Hidden Markov models (HMMs) [40]. Our work on HMM IE will solve the extraction redundancy issue in current HMM IE modeling on an entire document. To this end, we propose a segment-based HMM IE approach, in which a segment retrieval step is included to identify extraction-related segments from the entire document.

Our segment-based IE modeling is actually a general framework. In addition to HMM IE applicability, the same segment-based IE framework applies to other IE models in which the extraction is performed by sequential state labeling. To improve the system's adaptability to the situation when the labeled texts are limited, we will extend our segment-based HMM IE modeling to semi-supervised learning using a modified version of the multi-view Co-EM learning strategy.

Motivated by the need to choose term weighting related system design choices in segment retrieval in our segment-based HMM IE, we also investigate the use of information theoretic principles as tools to analyze the term vector models employed in IE and information retrieval (IR). Thus far, a series of theoretical analyses [23] show that the information theoretic principles provide a good framework to help make related design decisions in term vector models with sound theoretical justifications.

2.2 Terminology extraction

Acquisition of correct terminology (names of drugs, treatments, medical conditions) is a key research component. We consider two approaches: (1) "n-gram based approach"—classification approaches based on character and word n-grams are successfully used in terminology extraction from biomedical texts ([28], cf. [10,12,13,34]). Using character and word ngrams, we identify sets of salient features that can be used to annotate important domain-specific terms in abstracts of biomedical papers (GENIA corpus [29,45,46,50]);[4] and (2) "learned gazetteer approach"—named entity recognition (NER) is a key process focusing on extraction of proper names and natural kind terms from text [61]. For example, in the sentence: "*Examples of commonly used analgesics are over-the-counter medicines like Anacin (acetaminophen)*", the proper name "Anacin" would be recognized as a "drug name" and the term "acetaminophen" would be recognized as a "chemical compound". A NER system could serve as a text preprocessing step where names of drugs, chemical compounds and medical conditions are recognized. Most drug names are non-ambiguous (they do not intersect with common nouns), allowing NER to perform with high accuracy.

Many approaches to acquire correct terminology exist [10,12,13,28,34,44,61]; one is the use of semi-conditional random fields [53]. Semi-CRFs are a conditionally trained version of semi-Markov chains. Text segments often have a clear intuitive meaning, e.g., NER, in which a segment corresponds to an extracted entity; however, similar arguments might be made for several other tasks, such as gene-finding [12,34]. In NER, a semi-Markov formulation allows one to easily construct entity-level features (such as "entity length"). Preprocessed texts with NER meta-information are at the cornerstone of advanced information extraction tasks like question answering, text mining and semantic information retrieval [10,12,13,34,61]. Information distillation tasks include linking entities together (e.g., the entity "Tylenol" is frequently near the medical condition "headache"), clustering/classifying texts (e.g., group text by drug category) and mining large collections of texts can reveal unsuspected relations/interactions between entities.

Computational understanding of natural language is a perpetual problem; a complete idealized solution is not feasible, and many working solutions are tailored toward specific

[4] Terminology extraction from medical documents is an important foundational step in building medical ontologies for a sub-domain, and to find best evidence for best practice recommendations. Use of character n-grams for medical term extraction was also proposed in [24], where medical domain morphemes are detected using character 4-grams. Hishiki et al. [24] argues that such a method is necessary since we cannot completely rely on the existing medical dictionaries. Medical terminology is virtually unbounded, and new terms are constantly created.

applications. The problem has drawn attention due to requirements of shallow semantic annotation for question answering [30], better information retrieval, and semantic web forms of XML-based annotation [54]. Our approach harmonizes low-level statistical methods based on n-grams, and higher-level unification-based methods.

Using extracted and annotated terminological terms; we build composite semantic structures using regular expression rewriting and unification-based grammars [1,30,31]. Annotation is driven by the question category, e.g., the typical factoids annotated are time expressions, people and organization names, geographical locations, important events and properties such as when somebody died. We tailor our annotation to a health care domain, moving from a general but shallow scope to a more narrow deeper scope. We rely on our unification-based methodology with relaxed-unification[5] for semantic annotation. Semantic annotation at this level can be used to support our main IE task, and then use IE results to provide more complex semantic nuggets, which can be used in automatic inference. The use of soft computing techniques in data mining with regular expression and unification-based techniques provides wider functionality in a system for finding the best evidence in health care documents. Regular expression-based techniques are successful as robust methods for several natural language tasks [3,51].

2.3 Beyond factoids: discovering interesting patterns from health care databases

Digital patient historical records and progress notes contain valuable information about the patient medical care. Such information can be used to discover care patterns received by different patient types, to identify drug misuse, to make effective treatment method recommendations, and to predict health care expense for a particular patient type. After patient history extraction from narrative text, we obtain a sequence of medical events for each patient. A medical event can be a test result (e.g., an electrocardiographic result), a symptom experienced by the patient (such as pain and its location), a diagnosis by a doctor, a drug taken by the patient, the medication effect, and so on. Thus, after the IE process, a database of medical event sequences, one for each patient, is obtained. From this database, we apply data mining techniques to discover relationships among medical events, identify groups of patients that have similar medical event sequences and discover common patterns within each group.

2.3.1 Adaptive medical event sequence clustering with Markov models

We construct a probabilistic framework for clustering medical event sequences (and patients), see [39]. A data point in a sequence can be multivariate clinic feature vectors of various dimensionalities. In particular, we develop a mixture of Markov model-based framework[6] [39] for clustering such heterogeneous clinic data.[7] This form of clustering provides advantages over traditional non-probabilistic approaches. First, the model-based approach provides a general description of each identified cluster, so that patients are naturally profiled when

[5] Relaxed unification addresses inconsistencies in natural language, hence instead of simply failing because of insufficient rules coverage, a system can choose the most probable among apparently inconsistent results, and proceed.

[6] Note that the reason for using a mixture model with multiple components instead of a single Markov model is to distinguish different groups of sequences. Each component represents the characteristics of a group of sequences. A single Markov model would represent the general characteristics of all sequences.

[7] Model-based cluster analysis places it on a principled statistical support [15,38,56] based on probability models in which objects are assumed to follow a finite mixture of probability distributions such that each component distribution stands for a unique cluster.

clustering the sequences of clinic feature vector, and specific information about individual patients is preserved. Second, statistical models allow clustering uncertainties in component members, which is important for those objects close to cluster boundaries. In practice, a patient may suffer from different syndromes at the same time. It is thus of critical importance to assign the patient to more than one group, and give appropriate therapies accordingly, for a example see [52].

We will also investigate the possibility of building an adaptive clustering system, which can incrementally adapt the mixture of Markov models with new available data.[8] The learned mixture of Markov models can be used to make recommendations. First, given a "new" sequence of events, the mixture of Markov models can be used to predict next possible events with high probabilities. If this new event sequence is different from the predicted ones, our system can issue an alert for possible drug misuse or mistreatment. Second, by analyzing the learned mixture of Markov models, we can identify conditions under which a drug is effective for a disease and conditions under which it is not. The benefit of using Markov models is that the ordering of the events can be considered. Third, with the clustering result, we can identify the outliers in the patient historical records so that previous medical errors may be identified.

2.3.2 Direct and indirect association mining

For the health care applications, association rules can reveal the correlation relationship between two variables or two groups of variables, such as the correlation between the drugs used for diabetes patients and the effect of these drugs. To learn association rules, we examine three themes. First, although many algorithms have been developed to mine association rules (e.g., [4]), there remains a major problem. That is, not all of the generated association rules are interesting or useful [25]. This problem is referred to as the rule quality problem. To address this problem, measures of interestingness are used to assess the significance of a rule. Second, we investigate whether indirect associations can be used in health care applications. An indirect association is a special type of negative association that relates two objects via a mediator. The two objects in an indirect association occur frequently together with the mediator. For example, an indirect association rule can be "the combination of drug A and drug B is not effective for a disease, but each of them is effective with the use of drug C". We use our algorithm, HImine [57], for mining indirect association rules from data. Finally, we design a method that uses interesting association rules, interesting correlations and indirect associations to find interactions of multiple drugs from medical event sequences. Our objective is to build a robust recommendation system that suggests the most cost-effective and positive treatment plan for patients.

2.4 Validating evidence for recommendation (MSBNs)

Sometimes decisions on recommendations are made by different sources of recommendations. For example, the first topic of COMPUS is to identify recommendations for best practice in the prescribing and use of proton pump inhibitors.[9] Proton pump inhibitors are used to

[8] In clinic applications, the new clinic data may arrive continuously; ignoring those new data entirely without adjusting the model would adversely affect the system performance. Nonetheless, it could be cost prohibitive to rebuild the model using all the data available up to date. Therefore, a more efficient and scalable approach is desirable.

[9] Presently COMPUS gathers recommendations and supporting evidence from different guideline documents according to a well-established methodology. Most evidence was generated under controlled ideal conditions. Such evidence is presented to a group of experts who review the evidence and make best practice

heal stomach and duodenal ulcers including stomach ulcers caused by taking nonsteroidal anti-inflammatory drugs. Proton pump inhibitors are also the drugs of first choice for a rare condition called Zollinger–Ellison syndrome. The question that naturally arises is this: “Are recommendations made first and then corroborating evidence found or vice versa?”, or, more to the point, how do we tell “good evidence” from “bad evidence”?

We use probabilistic inference [49] to evaluate evidence and identify the best recommendations because evidential information is sometimes inconsistent and changes over time. There are also a number of other variables, as well in the complex process of making recommendations, including: the different literature reviewed; reviewers differ; the same literature can be assessed differently; decision-making will vary between experts; different outcome measures can be used; and influence from competing interests. A probabilistic model should be of value to examine and evaluate these numerous uncertain factors. Since decisions on recommendations for clinical practice may involve multiple parties and knowledge on different medical fields, an object-oriented multi-agent paradigm should be of value. Multiply sectioned Bayesian networks (MSBNs) have been successfully applied in static domains under a cooperative multi-agent paradigm [59]. In an MSBN, each section (agent) holds a partial perspective of a large domain and has access to local evidence. Global evidence can be obtained by communicating with other sections (agents) which can then update beliefs with local evidence and global information from other sections. Nevertheless, evidential information on which to base recommendations may evolve; we propose to apply dynamic multi-agent probabilistic inference to the problem. Probabilistic reasoning in dynamic multi-agent domains involves several issues [5]. Intuitively, observation on recent state plays a more important role in the reasoning of the current state than remote historic information. Based on the intuition, we model an entire domain for a time period into an MSBN and then reason about the state of the dynamic domain by observing as much relevant evidence as possible. We have proposed algorithms for efficient observation and computation for multi-agent probabilistic inference using MSBNs [5].

Eventually, the MSBN model can augment the expert panel to help evaluate evidence and identify the best recommendations. MSBNs make it possible to represent and handle all sorts of uncertain domain knowledge from different parties in a modular and/or distributed way.

3 Examining selected techniques more closely: naïve Bayes IE

We have observed problems in previous work on applying naive Bayes for IE in their formulation of IE as a maximum a posteriori (MAP) learning problem, ignoring certain aspects inappropriately. In view of this formulation problem, we first illustrate how to formulate an information extraction task to a Bayesian learning problem. Then based on the formulation, we show that various naive Bayes models, from the simplest one to more complicated models, can be induced by making different assumptions on probability estimation with regard to its dependence to contextual information.

Recall Bayes’ theorem $P(h|D) = P(D|h)P(h)/P(D)$. Given h from some hypothesis space H, $P(h)$ denotes the prior probability of h. $P(D)$ denotes the prior likelihood of the training data D being observed (i.e., given no knowledge about which hypothesis holds), and $P(D|h)$ denotes the likelihood of observing data D given hypothesis h holds. The left side of the formula $P(h|D)$ is called posterior probability of h given the observed data D, which

Footnote 9 continued
recommendations based on the evidence and their personal clinical experience. Perhaps the evidence itself might have a more prominent place in these recommendations?

is the probability that we usually intend to obtain in a machine-learning problem. Hence, the Bayes' theorem provides us a way to calculate the posterior probability $P(h|D)$ from the prior probability $P(h)$, together with data likelihood $P(D)$ and $P(D|h)$.

Many machine-learning tasks can be formulated as a search problem, which is to find the most probable hypothesis $h \in H$ given the observed data D. Such h is also called a MAP hypothesis. More formally,

$$\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{\text{argmax}}\, P(h|D) && (1)\\ &= \underset{h\in H}{\text{argmax}}\, P(D|h)P(h)/P(D) && (2)\\ &= \underset{h\in H}{\text{argmax}}\, P(D|h)P(h) && (3)\end{aligned}$$

In (4), the term $P(D)$ can be dropped from the formula because it is a constant independent of specific hypothesis h. It is also worth mentioning that the observed data D in (4) should refer to the same data D as in previous Eqs. (1 and 2).

3.1 Previous problems in applying naive Bayes models to IE

In [18], the naive Bayes learner has been applied to information extraction by adapting text classification modeling methods. In order to formulate the identification of slot fillers in text as a machine-learning problem, the hypothesis used in [18] takes the form of "the text fragment starting at token i with length of k tokens is the slot filler", denoted as $H_{i,k}$, where i takes any positive integer from 1 to the length of the entire document, and k is a positive number less than a specified threshold for the context window size. Then, (3) is used to search for most probable slot filler from all possible slot fillers in text.

There is a major difference between text classification and information extraction. To classify a document D, all tokens appearing in the document are considered as potential classification features. $P(D|h)$ gives the estimation for the likelihood of observing all tokens in document D if the document is known to belong to a particular class. However, in order to identify a slot filler in D, we are usually only concerned with the tokens that occur in and around the text fragment to which $H_{i,k}$ corresponds, as indicators for filler identification. To form our estimate of the likelihood of a filler candidate $H_{i,k}$, we take into account a fixed-size context window on either side of the hypothesized text fragment. Given a context window size m, the calculation of the filler probability for each $H_{i,k}$ not only considered all the tokens within the corresponding text fragment, but also includes the m tokens on either side of the hypothesized filler. *This difference implies that the naive Bayes modeling in IE should be different from that in text classification.*

The problem of the naive Bayes IE learner in [18] rests on the estimation of $P(D|H_{i,k})$, the likelihood of observing the document D given the hypothesis $H_{i,k}$. The calculation of this likelihood takes the product of the individual term probabilities only for the tokens that appear within a fixed-size context window around the slot filler corresponding to $H_{i,k}$, while all the other "nonlocal" tokens are ignored. Therefore, the calculated result indicates the likelihood of observing a small part of document D (within the specified context window of the filler), rather than the likelihood of the entire document given the hypothesis.[10]

[10] The data in document D is divided into two parts with regards to $H_{i,k}$, D_{context} denoting all the tokens appearing within the context windows around the corresponding filler, and $D_{\text{non-context}}$ denoting all the other tokens appearing in the document.

3.2 Naive Bayes modeling for IE

We base our naive Bayes modeling on the general formula (4).

$$\underset{Hi,k \in H}{\operatorname{argmax}} P(H_{i,k}|D) = \underset{Hi,k \in H}{\operatorname{argmax}} \frac{P(D_{\text{context}}|H_{i,k})P(H_{i,k})}{P(D_{\text{context}})} \tag{4}$$

In order to design a concrete naive Bayes learner for information extraction, details on how to estimate each probability in (4) need to be specified. We describe methods for estimating these probabilities.

3.2.1 Estimate of P(Dcontext)

The estimation of the prior data likelihood P(Dcontext) is straightforward, as follows.

$$P(D_{\text{context}}) = \prod_{t \in D_{\text{context}}} P(t)$$

where $P(t)$ of term t can be estimated from the training corpus as:

$$\text{\# occurrences of term } t \text{ in corpus/\# all term occurrences in corpus}$$

3.2.2 Estimate of $P(H_{i,k})$

To estimate the prior probability of a hypothesis, assume that every competitive hypothesis $H_{i,k}$ is equally probable a priori. In this case, $P(H_{i,k})$ in (4) is omitted since it is a constant independent of different $H_{i,k}s$.

Another seemingly more reasonable estimate of the prior probability of hypothesis $H_{i,k}$ is the expectation of the chances, based on the training corpus, that a slot filler would occur starting at the ith token and be of length of k tokens. We assume that these two filler factors (i.e., the starting position and the length) are independent. So, we can calculate the prior probability of a hypothesis by estimating the probabilities of filler starting position and the probabilities of filler length separately as following.

$$P\left(H_{i,k}\right) = P\left(\text{position} = i\right) P\left(\text{length} = k\right)$$

3.2.3 Estimate of $P(Dcontext|H_{i,k})$

We considered several ways to estimate the data likelihood of the related context given hypothesis. The first decision to make is which part of text should be regarded as contexts of the hypothesized slot filler. The simplest way is to only consider the individual tokens appearing in the field itself.

3.2.3.1 Tokens-in-filler as direct contextual indicators

$$P(D_{\text{context}}|H_{i,k}) = \prod_{j=i,\ldots,i+k-1} P(t_j|H_{i,k})$$

where t_j is the jth token in document D. The individual term probabilities can be readily estimated from the provided training corpus:

$$\begin{aligned} P\left(t|H_{i,k}\right) &= \Pr\{\text{term}\, t \text{ appearing in fillers}\} \\ &= \#\text{ occurrences of term}\, t \text{ in fillers}/\#\text{ occurrences of all terms in fillers} \end{aligned}$$

The Bayes Basic IE learner is implemented by using this estimating method.

3.2.3.2 Tokens-in-context and tokens-in-filler as same kind of contextual indicators Since an IE hypothesis is meant to identify related text fragments, it is natural to take into account the tokens close to the hypothesized field to help determine relevance. For this purpose, a context window of fixed size is applied on either side of the filler. When we estimate the likelihood of filler, all tokens within the context windows, including tokens in the filler, will be considered in the calculation. Suppose, we set the window size to m tokens, then we use the following formulas to calculate the filler likelihood.

$$P(D_{\text{context}}|H_{i,k}) = \prod_{j=i-m;:::;i+k+m-1} P\left(t_j|H_{i,k}\right);$$

and

$$\begin{aligned} P\left(t|H_{i,k}\right) &= \Pr\{\text{term}\, t \text{ appearing in fillers or their contexts}\} \\ &= \#\text{ occurrences of term}\, t \text{ in fillers } + \#\text{ occurrences of term}\, t \text{ in fillers' contexts}/ \\ &\quad \#\text{ occurrences of all terms in fillers } + \#\text{ occurrences of all terms in} \\ &\quad \text{fillers}' \text{ contexts} \end{aligned}$$

In this case, the estimated individual term likelihood implies how likely the specific term would appear in the hypothesized filler or in the context windows on either side of the filler. All the tokens considered serve as the same kind of filler indicators regardless of their appearing positions as related to the filler.

The Bayes Context IE learner is implemented by using this estimating method.

3.2.3.3 Tokens-in-context and tokens-in-filler as different kinds of contextual indicators We can include some ordering information of tokens in the filler likelihood estimate by dividing the tokens in the area of concern into three different sets: one set of tokens, called pre-context tokens, that appear within the context window before the filler; one set of tokens, called in-filler tokens, that appear within the filler itself; and the other set of tokens, called post-context tokens, that appear in the context window after the filler. The counting method of probability estimation therefore needs to be adjusted accordingly by not only considering the occurrence of a token in the context window but also considering whether the token is a pre-context token, an in-filler token or a post-context token. Formally, the data likelihood estimate is computed as:

$$\begin{aligned} P\left(D_{\text{context}}|H_{i,k}\right) &= \prod_{j=i-m,\ldots;i-1} P_1\left(t_j|H_{i,k}\right) \prod_{j=i,\ldots,i+k-1} P_2\left(t_j|Hi_{i,k}\right) \\ &\quad \times \prod_{j=i,\ldots,i+k+m-1} P_3\left(t_j|H_{i,k}\right) \end{aligned} \tag{5}$$

where P_1, P_2 and P_3 are different probability distributions, and they are associated with pre-context tokens, in-filler tokens and post-context tokens, respectively. In the training stage, for each term that appears in the training data, we need to estimate these three probabilities associated with that term. So, the same term could provide different clues to a hypothesized filler according to the position it occurs in the text with regard to the filler. In detail, these three term probability distributions can be estimated during the training phrase as follows:

$$\begin{aligned}
P_1\left(t|H_{i,k}\right) &= \Pr\{\text{term t appearing in pre-contexts of fillers}\}\\
&= \#\text{ occurrences of term } t \text{ as pre}-\text{context tokens}/\\
&\quad \#\text{ occurrences of all terms as pre}-\text{context tokens}\\
P_2\left(t|H_{i,k}\right) &= \Pr\{\text{termtappearing within fillers}\}\\
&= \#\text{ occurrences of term } \quad t \quad \text{as in}-\text{filler tokens}/\#\text{ occurrences of all terms}\\
&\quad\ \text{as in}-\text{filler tokens}\\
P_3\left(t|H_{i,k}\right) &= \Pr\{\text{term } t \text{ appearing in post-contexts fillers}\}\\
&= \#\text{ occurrences of term } \quad t \quad \text{as post}-\text{context tokens}/\\
&\quad \#\text{ occurrences of all terms as post}-\text{context tokens}
\end{aligned}$$

The intuition of this modeling is that certain ordering might exist in the contextual tokens of slot fillers because of the syntactically organized written language. For fillers of a specific slot, there might exist some tokens appearing frequently right before the text fragments corresponding to the fillers. Similarly, some other tokens might usually occur right after the slot fillers, and another different set of tokens would often appear as part of the fillers themselves.

The Bayes ContextPro IE learner is implemented by using this estimating method.

3.2.3.4 Token positions in context also as contextual indicators To take this idea even further, we consider including token position information in context in the modeling. Thus, we no longer make assumptions that all tokens in the pre-context fragment or in the post-context fragment follow the same term probability distribution, i.e., P_1 or P_3 in formula (5). Within the fixed-size context window on either side of the hypothesized filler, the tokens at different position would follow different term distributions, which can be estimated from the training corpus separately. Thus, our estimate to the filler likelihood becomes:

$$\begin{aligned}
P\left(D_{\text{context}}|H_{i,k}\right) = &\prod_{j=i-m,\ldots,i-1} P_{1,i-j}\left(t_j|H_{i;k}\right) \prod_{j=i,\ldots,i+k-1} P_2\left(t_j|H_{i,k}\right)\\
\times &\prod_{j=i+k,\ldots,i+k+m-1} P_{3,j-i}\left(t_j|H_{i,k}\right);
\end{aligned}$$

where $P_1; l(l = 1, \ldots, m)$ is the term distribution of the lth tokens in the pre-context to the left of the hypothesized filler, and likewise, $P_{3,l}(l = 1, \ldots, m)$ is the term distribution for the lth tokens in the post-context to the right of the hypothesized filler. These probability distributions can then be estimated from the training corpus as follows.

$$
\begin{aligned}
P_{1,l}\left(t|H_{i,k}\right) &= \Pr\{\text{term } t \text{ appearing as } l\text{th token before fillers}\}\\
&= \#\text{ occurrences of term } t \text{ as } l\text{th tokens before fillers}/\\
&\quad \#\text{ occurrences of all terms as } l\text{th tokens before fillers}\\
P_{3,l}\left(t|H_{i,k}\right) &= \Pr\{\text{term } t \text{ appearing as } l\text{th token after fillers}\}\\
&= \#\text{ occurrences of term } t \text{ as } l\text{th tokens after fillers}/\\
&\quad \#\text{ occurrences of all terms as } l\text{th tokens after fillers}
\end{aligned}
$$

The Bayes Position IE learner is implemented by using this estimating method.

3.3 Implementation of naive Bayes IE learners

Based on the different naive Bayes modeling methods described above, we have implemented several naive Bayes IE learners, and evaluated their extraction performances on the well-known seminar announcement domain for purposes of comparison. The data set consists of a set of labeled seminar announcements provided with specification of four slots to be extracted: *location* (seminar location), *speaker* (the speaker of a seminar), *stime* (the seminar starting time) and *etime* (the seminar ending time).

In all these models, the approaches for estimating the prior hypothesis probability and the prior data (i.e., text fragment of interest in this case) likelihood are same as specified in [23]. They vary in their ways of estimating the data likelihood of the hypothesized slot filler.

1. *Bayes Basic IE learner* This naive Bayes learner only utilizes tokens within the hypothesized slot filler as indicators to calculated the filler likelihood. No contextual information or token ordering information is considered in this learner. It is the simplest learner among those we implemented in our experiments.
2. *Bayes Context IE learner* Contexts of the hypothesized filler are taken into consideration in this IE learner, by applying a fixed-size context window on either side of the text fragment corresponding to the hypothesized filler. The tokens appearing in the two context windows (i.e., pre-context tokens and post-context tokens) play the same role in being filler indicators as the tokens located within the text fragment corresponding to the hypothesized filler.
3. *Bayes ContextPro IE learner* In this learner, more contextual information of the hypothesized filler is considered compared to the Bayes Context IE learner. The tokens under consideration are divided into three kinds according to their position relative to the hypothesized filler: pre-context tokens for the tokens located in the fixed-size context window applied right before the text fragment corresponding to the hypothesis; in-filler tokens for all the tokens appearing within the hypothesized filler; and post-context tokens for the tokens whose positions are in the range of the context window on the right side of the hypothesized filler. These three kinds of tokens serve as three different kinds of filler indicators in this learner, and follow different term probability distributions which are obtained from the training corpus. However, for the tokens belonging to the same kind of tokens, their contributions to the overall filler likelihood are determined from the same term distributions associated to the pre-context tokens, the in-filler tokens, or the post-filler tokens.
4. *Bayes Position IE learner* This learner is most context-sensitive one among all these naive Bayes learners. In addition to differentiating three kinds of tokens associated to the hypothesized filler as in the Bayes ContextPro learner, the ordering of the tokens in

the two context windows is applied on either side of the text fragment corresponding to the hypothesized filler. Each location in the context window is treated as a different kind of filler indicator, and the contribution of the token at different locations to the estimated likelihood of the whole filler is determined by different term probability distributions which are obtained from the training set separately.

These four naive Bayes IE learners take more and more concrete contextual information in their estimation for the filler probability. The more complicated is the modeling, the more probabilities that need to be estimated in the naive Bayes IE learner during the training phase, from a fixed observed data.

3.3.1 Text preprocessing in naive Bayes IE

Very modest text preprocessing is needed for naive Bayes IE systems. Text must be tokenized before we perform term counts in the training set to estimate term distributions in a naive Bayes IE system. We considered two different text tokenizations. One way treats punctuation marks as token delimiters and removes them after the text is transferred to a sequence of tokens. The other way counts punctuation marks as separate tokens. Our experimental results show no significant difference on system performance using these two tokenization methods. Use of punctuation marks as tokens give slightly better performance for some slots of the naive Bayes' learners, but also cause a slight performance decrease for some learners on another slot. The experiment data reported in [23] all use a text tokenizer, which count punctuation.

3.3.2 Extraction threshold and context size

For naive Bayes IE learners, the system is usually able to return an extraction unless a probability threshold is enforced or the problem of zero probability propagation is too serious to find an extraction from all the filler candidates, which is rare even with no probability smoothing. Unless users of the IE system place much more priority on high extraction recall, we regard that a naive Bayes IE system should apply an extraction threshold to avoid spurious extractions. The use of an extraction threshold in naive Bayes IE systems can also provide us a method to adjust a trade-off between extraction precision and recall in the system overall performance depending on the user requirement on the specific extraction task.

To choose the extraction threshold for further extraction, we determine it from the training set as the least likelihood of all labeled fillers in the training data calculated using learned term probability distributions.

For the naive Bayes learners, which use contextual information around fillers, another system parameter that needs to be determined is context window size, which could be specified in advance as some reasonable number (such as 5 tokens for both pre-context window and post-context window size as our default setting). The slot-specific optimal window size can also be learned from a series of training runs of the process over the training set with a different window size setting within a given range (e.g., from 1 to 5 tokens), and select the setting which gives best performance on the training set.

3.3.3 Smoothing methods for naïve Bayes IE probability estimation

A manifest failure of the maximum likelihood estimation, which is used in the probability estimation in these naive Bayes learners, is that unseen events from the training data provided would be assigned with zero probability. This leads to inaccurate estimates since it underestimates the frequencies of the events, which are not seen in the training sample, and overestimates the events appearing in the sample.

Moreover, in the naive Bayes models, the estimated filler probability is basically the product of the probabilities of all the tokens appearing in corresponding contextual positions. The assignment of unseen events with zero probability would cause a zero propagation disaster, in which the probability calculation would assign any filler candidates containing tokens with zero probability with the minimal likelihood (i.e., maximum negative value). Hence, such filler candidates would be filtered out, even if there were other tokens existing in them showing strong filler indication. When the training data is very limited and the probability number to be estimated is very large, it is possible that there are unseen events associated with every filler candidate in the whole text of a seminar announcement to be extracted. In this case, there would be no extraction returned as the result.

It is not reasonable to assign zero probability to all unseen events when a probability distribution is estimated from a set of sample data. Smoothing (or "discounting") methods are used to address this problem, in which unseen events are assigned with a relatively small probability, and the probabilities of other observed events in the sample data would be adjusted accordingly in order to keep the total sum of the event probabilities to be one. In order to choose an appropriate probability smoothing method used in our naive Bayes IE learners, as well as to be used in other statistical IE learners, we investigated some commonly used smoothing techniques.

Laplace Smoothing A common smoothing method that has been applied in many statistical learners is Laplace's law which assigns a small part of the probability space to unseen events [42]. According to the Laplace's law, an event's probability is estimated from its occurrences in the sample data as follows.

$$P_{Lap}(e) = \frac{C(e) + 1}{C(\sum e) + |\sum e|} \tag{6}$$

where $C(e)$ is the number of occurrences of event e in the sample, $C(\sum e)$ is the total number of occurrences of all the events observed, and $|\sum e|$ stands for the number of different events observed in the sample (often referred to as the alphabet size). It is naturally called "adding one" smoothing, since the probability estimation from the observed event occurrence adds one occurrence to each event, including one assumed unseen event, and then smooths all probabilities by re-normalizing them over the entire alphabet. In this way, the unobserved event is assigned with a relatively small probability instead of zero.

Simple good-turning smoothing (SGT) This method [49] is a population frequency estimator, which adjusts observed term frequencies to estimate the real population term frequencies and assigns a probability to all unseen events according to the estimated population probability distribution from the sample.

The observed frequency distribution from the sample can be represented as a vector of (r, n_r) pairs, $r = 1, 2, \ldots$. In our naive Bayes models for IE, r are the observed term frequencies from the training data, and n_r refers to the number of different terms which occur with frequency r in the sample. For each r observed in sample, the Good-Turning methods give estimation for its real population frequency according to (7).

$$r^* = (r + 1)\frac{E(n_{r+1})}{E(n_r)} \tag{7}$$

$E(n_r)$ is the expected number of terms with frequency r.

For unseen events, an amount of probability P_0 is assigned to all these unseen events, as $P_0 = E(n_1)/N$ where N is the total number of term occurrences in the sample.

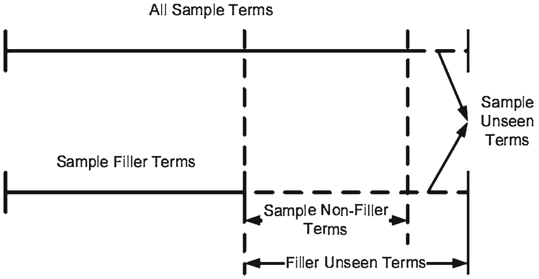


Fig. 1 Term smoothing in naive Bayes IE

3.3.4 Our probability assignment among unseen terms in naïve Bayes IE systems

The probability assigned for unseen events by SGT smoothing is a total probability, and how this amount of probability is assigned among all unseen events is a practical concern, which needs to be addressed appropriately when we apply a smoothing method to a specific probability estimation problem.

For the estimation of the probability $P(t)$ of term t in our naive Bayes models, it is reasonable to regard all unseen terms as one kind of special term and assign P_0 to the new terms encountered in the extraction. This is due to the fact that most of the terms in the alphabet would be seen in the training data set, with considerably less unseen terms encountered when a new text is input to the system for extraction.

However, the conditional probabilities $P(t|H)$ would have a much more serious data sparseness problem. Consider all terms, which could occur in the context of the fillers for a specific slot as the alphabet set of the population. Since labeled training texts provided are limited, only a small subset of the alphabet can be observed in the sample, and it is much more frequently we would encounter the unseen terms which do not occur in the filler segments in the sample when extraction on a new document is performed. We assume that the conditional probability distribution shares the same alphabet set as the prior probability distribution, that is all possible terms in the domain. The difference of these two distributions lies in that filler related conditional probability would exhibit more skewed distribution than all term distributions.

For the term probabilities associated with fillers, there are two kinds of unseen terms, which do not appear in the filler segments in the sample. One kind is those terms, which appear in other places in the sample but do not appear in the fillers; the other kind is sample unseen terms. Figure 1 shows these two kinds of filler unseen terms and their relationship to all term distributions. We call the first kind of filler unseen terms as sample non-filler terms, since they occur in the sample texts somewhere other than the places where any filler is located. We regard sample non-filler terms are good indicative terms that tell us that the text segments containing such sample non-filler terms are very unlikely to be a filler. Hence, we do not assign any unseen probability to sample non-filler terms, while all the unseen probability is assigned to the sample unseen terms.

For the term probabilities associated with terms in certain contextual positions, the related unseen terms can be arbitrary kinds, such as filler terms, context terms at other positions, or terms occurring outside the context windows around fillers. In this case, it is extremely difficult to design an appropriate assignment of total unseen probability among all these different kinds of unseen events. We assume a uniform distribution among all these unseen terms, and estimate the number of species for the unseen events in the contextual term probability distribution as the size difference between that specific term distribution and all term distribution, then assign P_0 uniformly among all these species.

3.3.5 Experimental results and performance comparison to related work

In [23], we discuss traditional performance metrics (precision-recall and accuracy-coverage metrics) and better performance evaluation metrics for IE are presented. We regard a more appropriate performance measure in IE should be *accuracy-recall*. However, we report most of our experimental results in *precision-recall* pairs, as well as the $F1$ measure for an overall performance indicator. We do so in order to make meaningful comparison with other IE systems. Our experimental results for the four IE learners and performance comparison to related work can be found in "Appendix A".

3.3.6 In summary

We presented a formal naive Bayes modeling for IE problems, and induced advanced naïve Bayes models to include certain contextual features. Our formulation corrected the problem of ignorance of context prior data likelihood that existed in previous work. The filler probability calculation formula obtained directly from our formulation, with no need for any heuristic modification or adjustment, has been shown performed well. When more and more detailed contextual information is added in the modeling, the resulting model would become more complex in which much more probability numbers need to be estimated from the limited training examples. In order to alleviate the data sparseness problem in the training of advanced naive Bayes models, we applied smoothing methods for more stable probability estimation, and proposed approaches on how to estimate the number of species for unseen events and on how to assign total unseen probability appropriately among all unseen terms. Our experimental results show that a well-designed smoothing method is critical to the system's robustness, particularly for more complex statistical models. We choose to first work on Naive Bayes modeling in IE is not only because of its simplicity, but also because, surprisingly, the *existing work on naive Bayes IE is limited and needs to be improved to give a better estimate on the IE learning capability of this popular statistical model*. In addition, our current naive Bayes IE systems are purely adaptive systems, since virtually no settings need to be adjusted to apply the same naive Bayes IE learner to a different domain, and no additional text preprocessing is required either.

Another model, which has been proposed to apply to the problem of information extraction, is the hidden Markov model (HMM). We discuss HMMs next.

4 Examining selected techniques more closely: segment-based HMMs for IE

We implement two document-based HMM IE systems using two different kinds of HMM structures. HMM Basic uses the basic HMM structure as shown in Fig. 2 which only has two states, i.e., "Filler" state and "Non-filler" state. HMM Context uses the structure that is similar to the one specified in [20], which contains a limited number of pre-context and post-context states, as well as more than one filler paths. An example of HMM Context model structure is shown in Fig. 3, in which the number of pre-context states, post-context states, and the number of parallel filler paths are all set to 4, as the default parameters.

HMM Context consists of the following four kinds of states besides the special "start" and "end" states.

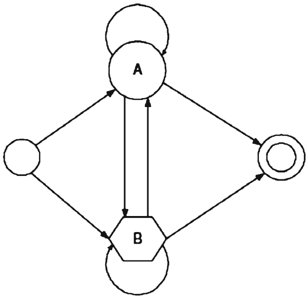


Fig. 2 A basic HMM for information extraction

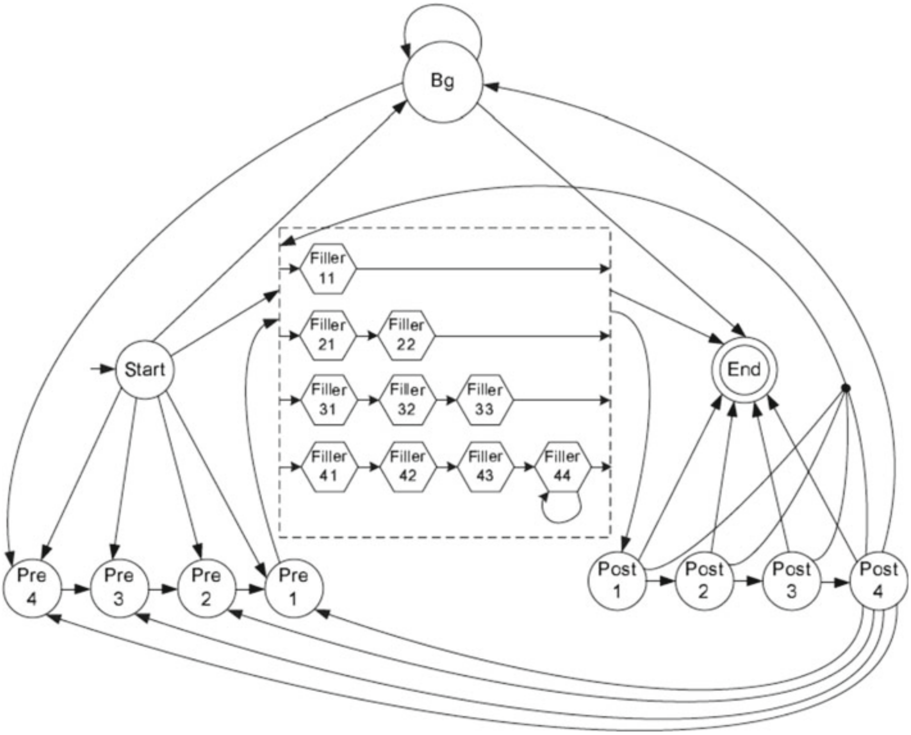


Fig. 3 An example of HMM context structure

- Filler states—this state corresponds to the event that slot fillers occur.
- Background state—this state corresponds to the event that neither slot fillers nor contexts of slot fillers occur.
- Pre-context states Pre_4; Pre_3; Pre_2; Pre_1—states correspond to the events of context tokens occurring before the fillers at the specific positions relative to the fillers, respectively.
- Post-context states $Post_1$; $Post_2$; $Post_3$; $Post_4$—states correspond to the events of context tokens occurring after the fillers at the specific positions relative to the fillers, respectively.

Our HMM Context structure differs from the one used in [20,40] in that we add the transitions from the last post-context state to every pre-context state as well as every first filler state. This handles the situation where two filler occurrences in the document are too close to each which makes the text segment between these two fillers shorter than the sum of the pre-context and post-context sizes.

4.1 Smoothing in HMM IE systems

There are many probabilities that need to be estimated to train an HMM for information extraction from a limited number of labeled documents. Compared to naive Bayes IE systems, the data sparseness problem in probabilistic learning is even more severe in HMM training in IE, especially for the more advanced HMM Context structure. Since the emission vocabulary is usually large with respect to the number of training examples, maximum likelihood estimation of emission probabilities will lead to inappropriate zero probabilities for many words in the alphabet.

Based on our previous SGT smoothing experience, with unseen species estimation exhibited in our naive Bayes IE systems, we apply the same smoothing method to our HMM IE systems to alleviate data sparseness problems in HMM training. In particular, the emission probability distribution for each state is smoothed using the SGT method, and the number of unseen emission terms is estimated for each state before assigning the total unseen probability obtained from SGT smoothing among all unseen events.

The data sparseness problem in probability estimation for HMM models has been addressed to some extent in previous HMM-based IE systems (e.g., [20]). Smoothing methods such as absolute discounting, a variation of the Laplace smoothing, have been used for this purpose. Moreover, Freitag et al. [20] uses a "shrinkage" technique for estimating word emission probabilities of HMMs in the face of sparse training data. It first defines a shrinkage topology over HMM states, then learn mixture weights for producing interpolated emission probabilities by using "held-out" data from labeled data.

Experimental results on document HMM IE are presented in "Appendix B".

4.2 Issues with document-based HMM IE evaluation

Previous work on applying HMMs to IE modeled the entire document as one long observation sequence emitted from the HMM. The extracted fillers are identified by any part of the sequence in which tokens in it are labeled as one of the filler state. The commonly used structure of HMMs in IE allows multiple passes through the filler states. Thus, it is possible for the labeled state sequences presenting multiple filler extractions. Performance reports in previous research regarding document-wise extraction evaluation are unknown, in which single extraction for each slot is enforced for each document, and how exactly a correct extraction for one document is defined in HMM IE evaluation. We consider correct extraction for a document either by any of the token segments that pass the filler states being a correct extraction, or by all the token segments that pass the filler states being correct extractions (i.e., an exactly state sequence matching with the original labeled HMM state sequence for that document). More likely, the former correctness criterion was used in evaluating the HMM IE systems in the related work. So, we evaluate our document HMM IE systems using this same criterion.

Although it is reasonable to regard a document being extracted correctly by any one of the found slot fillers from the labeled state sequences being a correct filler, certain issues exist when the document HMM IE system issues multiple extractions for the same slot for one document. For example, when multiple slot fillers are issued for one document, it is possible that some of them are not correct extractions. In this case, such document-wise extraction performance evaluation alone would not be sufficient to measure the HMM IE

Table 1 $F1$/redundancy in document HMM IE on SA domain

IE learner	Location		Speaker		stime		etime	
	$F1$	R	$F1$	R	$F1$	R	$F1$	R
Doc HMM Basic	0.636	0.046	0.142	0.136	0.607	0.635	0.557	0.450
Doc HMM Context	0.822	0.054	0.714	0.095	1.000	0.131	0.949	0.063

system. Document HMM IE modeling cannot provide any criteria to select one most likely filler from the ones identified by the state sequence matching over the whole document. For the template-filling IE problem that is of interest to us, the ideal extraction result is one slot filler per document. Otherwise, further post-processing would be required to choose one extraction, from the possibly extracted multiple fillers by a document HMM IE system, for filling in the slot template for that document.

4.3 Document extraction redundancy in HMM IE

We introduce another performance measure, *document extraction redundancy*, Definition 1, to be used with document-wise extraction correctness criteria to enable a more complete extraction performance evaluation in a HMM-based IE system.

Definition 1 Document extraction redundancy *is defined over the documents which contain correct extraction(s), as the ratio of the* incorrectly *extracted fillers to all returned fillers from the document HMM IE system.*

For example, when the document HMM IE system issues more than one slot extraction for a document, if all the issued extractions are correct ones, the extraction redundancy for that document is 0. Among the all issued extractions, the more there are incorrect ones, the closer the extraction redundancy for that document is to 1.[11]

We have observed that extraction redundancy exists in the document HMM IE systems whose experiments are reported in "Appendix B". We calculate the average document extraction redundancy over all the documents that are judged as correctly extracted, and the evaluation results on the document extraction redundancy (shown in the R column) for both Doc HMM Basic and Doc HMM Context systems are listed in Table 1, paired with their corresponding $F1$ scores from document-wise extraction evaluation.

The average extraction redundancy with Doc HMM Basic is worse than Doc HMM Context, particularly for the two time-related slots. Generally speaking, the HMM IE systems based on document modeling have certain extraction redundancies for any slot in this domain, and in some cases such as for "speaker" and "stime", the average extraction redundancy is by all means not negligible.

In order to make the IE system produce the ideal "one slot extraction per document," we propose a segment-based HMM IE framework in the following subsection, aiming at dramatically reducing the document extraction redundancy and making the resulting IE system output extraction results to the template filling IE task with the least post-processing requirement.

[11] It can never be 1 according to our definition, since the extraction redundancy is only defined over documents containing at lease one correct extraction.

4.4 Segment-based HMM IE modeling

To solve the inherent problem of having nonzero extraction redundancy in document HMM IE systems, we propose a segment-based HMM IE modeling method, in which an HMM is used to model only "extraction-relevant" part of texts, not entire documents. We refer to this modeling as segment HMM IE.

Our segment-based HMM IE modeling requires that each document in the IE domain can be segmented into two kinds of text fragments: either text segments that are related to the extraction or ones that are not related to the extraction.

Generally speaking, any existing text classification and/or text retrieval techniques can be adapted to identify slot-specific extraction-relevant text segments from the entire document. In order to train a text classifier to serve our text filtering purpose, we first prepare the training data by automatically "labeling" each segment in the text as either "extraction-relevant" or "extraction-irrelevant". If there is a slot filler appearing in the sentence, the sentence would be labeled as a positive example, and all other sentences, which do not contain any slot filler, would be labeled as negative examples. Then, a text segment classifier is trained from these labeled sentences to learn segment classification knowledge with respect to a specific slot to be extracted. The obtained segment classifier is then applied to unseen texts to filter "extraction-irrelevant" segments from the entire document, and feed only the "extraction-relevant" ones to the HMM in the next step for filler extraction.[12]

4.5 Segment-based HMM IE modeling: the procedure

By imposing an extraction-relevance text segment retrieval in the segment HMM IE modeling, we perform an extraction task using an HMM as two successive sub-tasks.

Step 1: Identify from entire documents the text segments, which are relevant to a specific slot extraction. In other words, the entire document is filtered by locating text segments (e.g., a sentence in the text) that might contain the extraction.

Step 2: Extraction is performed by applying the segment HMM to only the extraction-relevant text segments obtained from the first step. Each retrieved segment is labeled with a best state sequence give the HMM, and all segments are ranked by their normalized likelihood of its best state sequence. The fillers labeled by the top ranked segment are returned as the extraction.

Since it is usual for more than one segment have been retrieved at Step 1, these segments need to compete at step 2 for issuing extraction(s) from their best state sequence found with regard to the HMM used for extraction with parameters λ. For each segment s with token length of n, its normalized best state sequence likelihood is defined as follows.[13]

$$l(s) = \log(\max_{\mathrm{allQ}} P(Q, s|\lambda)) \times \frac{1}{n}. \tag{8}$$

Segments are then ranked according to $l(s)$, and the top one is selected and the extraction is identified from the labeled state sequence by the segment HMM.

[12] It is worth mentioning that the unit of the extraction-relevant text segments is definable according to the nature of the texts to be extracted. For most texts, one sentence in the text can be regarded as a text segment. For some texts that lack grammar properties, in which sentence boundaries are hard to identify, we can define the "extraction-relevant" text segments be the part of text that are within a context window of the filler occurrences. Depending on the nature of the segment retrieval method used for identify extraction-relevant segments, text segmentation would be or not required before segment retrieval is performed.

[13] Given the observation sequence O and the model λ, how to choose the state sequence Q that best explains O?

This proposed two-step HMM-based extraction procedure requires training of the IE models. First, we need to learn an extraction-relevance segment retrieval system from the labeled texts, which will be described in detail in Sect. 4.6. Then, an HMM is trained for each slot extraction by only using the extraction-relevant text segments rather than entire documents.

By limiting the HMM training on much smaller part of texts, basically including the fillers and their surrounding contexts, the alphabet size of all emission symbols associated with the HMM would be significantly reduced. Compared to the common document-based HMM IE modeling, our proposed segment-based HMM IE modeling also eases the HMM training difficulty caused by the data sparseness problem since we are working on a smaller alphabet.

4.6 Extraction-relevant segment retrieval

Our purpose of segment retrieval is to identify, from the entire document, the part of text (i.e., a small set of text segments) that is related to a particular extraction task. These retrieved extraction-relevant text segments are then treated as input to the IE learner or extractor in the next step for identifying the exact occurrences of the slot fillers within these text segments.

The goal of a well-designed segment retrieval system for our purpose is not solely to achieve good overall classification accuracy or retrieval precision, but a segment retrieval system which is easily adjustable to retrieve almost all the extraction-relevant segments from each document as well as to get as few irrelevant ones as possible. For our segment retrieval system, being able to achieve high retrieval recall is a goal with higher priority than the overall system retrieval performance (such as $F1$ measure used to evaluate retrieval precision and recall at the same time). The reason is obvious: we do not want to miss any possible relevant segments during the text filtering, the first step of the entire extraction process.

In [23], two-segment retrieval algorithms for identifying extraction-relevant segments from documents, and their evaluation as stand-alone segment retrieval systems are presented which meet our requirements. Both of the two-segment retrieval methods require the documents be segmented into small pieces before training the segment retrieval systems from labeled texts. One way to do this is to segment the text into sentences, as we implemented our text segmentation in our experiments [23].

4.7 In summary

In current HMM-based IE systems, an HMM are used to model at the document level, which causes certain redundancy in the extraction. We propose a segment-based HMM IE modeling method in order to achieve zero redundancy extraction using HMMs. In our segment, HMM IE approach, a segment retrieval step is applied first so that the HMM extractor identifies fillers from a smaller set of "extraction-relevant" segments. Our system's document-wise retrieval performance can give us greater insight into the goodness of a particular segment retrieval method to be used in our segment HMM framework. The resulting segment IE systems not only have achieved nearly zero extraction redundancy, but also have shown improvement on the overall extraction performance.

We regard that, for the kind of IE problems that are of interest, i.e., template-filling with sparse extraction (i.e., filler occurrences relatively distant to each in the document), it is more reasonable to perform extraction by HMM state labeling on segments, rather than on the entire document. When the observation sequence to be labeled becomes longer, finding the best single state sequence for it would become a more difficult task. The effect of changing a small part in a very long state sequence would not be as obvious, with regard to the state path probability calculation, as changing the same subsequence in a much shorter state sequence.

In fact, this perspective not only applies in HMM IE modeling, but also applies to any IE modeling in which extraction is performed by sequential state labeling.

Segment retrieval for extraction is an important step in segment HMM IE, since it filters some irrelevant segments from the document, leaving fewer segments to complete in the following steps for issuing extractions. The HMM for extraction is learned from extraction-relevant segments rather than on all segments, which enables the HMM to work on a much smaller alphabet set.

A future goal for segment HMM IE is to design segment retrieval methods, which do not require document segmentation before retrieval, hence avoiding the possibility of early-stage errors, introduced from the text segmentation step. A promising idea is to adapt our naive Bayes IE to perform redundant extractions directly on a document to retrieve filler-containing text segments for a segment HMM IE system.

5 Examining selected techniques more closely: an example of finding evidence from association rule algorithms

We demonstrate evidence discovery from individual encounters using an association rule algorithm. This experiment demonstrates our initial trial of traditional machine-learning and artificial intelligence techniques, such as rough sets theory, applied to find evidence toward evidence-based medicine practice.

In this section, we demonstrate how to select relevant attributes for rule discovery in the medical data set (input), then select important rules from the set of generated rules as output for deployment as evidence for evaluation in the overall decision support system [58].

Rough sets theory is commonly used for attribute selection in the decision-making processes. Efforts into applying rough sets theory to knowledge discovery in databases have focused on decision-making, data analysis, discovering and characterizing the interdata relationships, and discovering interesting patterns. The decision table consists of condition attributes and decision attributes. A reduct, see “Appendix C”, is a subset of condition attributes that can represent the entire data set. Traditionally, reduct generation is designed to extract important condition attributes from a decision table. By considering fewer attributes, the decision-making process will become more efficient, cf. [60]. Association rule algorithms are well known for discovering associations, e.g., shopping behaviors among transaction data. One of the main problems for association rule generation is that the number of rules generated is generally quite large [43], thus it is very difficult to evaluate and rank these rules. In order to solve this problem, many novel approaches have been developed to extract more interesting rules [35,47]. Rule templates [32] as one example of rule interestingness measures, are often applied to extract appropriate rules toward certain applications. Rule templates have proven useful in decision-making, recommender systems, and other applications.

Little effort to date has been made on applying rough sets theory to association rules generation. Since rough sets can be used to determine whether there is redundant information and the rough sets method can help generate representative attributes, we expect fewer rules will be generated based on fewer attributes, and the rules will be as significant as those generated without using the rough sets approach. After a reduct is generated, a rule based on this reduct is generated in the form such that the antecedents of a rule is from the value of condition attributes in the reduct set, and the consequent of a rule is from the value of decision attributes from the original data set. Association rules generation also returns rules with certain support and confidence.

Table 2 Reduct and rules for Iris data set

Reducts	Rule sets
{sepalLength, sepalWidth, petalLength}	{sepalLength4.4→setosa, sepalWidth2.9→ versicolor, petalLength1.9→setosa, …}
{sepalWidth, petalLength, petalWidth}	{sepalWidth2.9→versicolor, petalLength1.9→ setosa, petalWidth1.1→versicolor, …}
{sepalLength, petalLength, petalWidth}	{sepalLength4.4→setosa, petalLength1.9→ setosa, petalWidth1.1→versicolor, …}
{sepalLength, sepalWidth, petalWidth}	{sepalLength4.4→setosa, sepalWidth2.9→ versicolor, petalWidth1.1→versicolor, …}

An association rule algorithm can be used to extract rules from the decision table as well. We are interested in using rough sets theory to facilitate this type of association rule generation. We focus on how to use rough sets theory to discover important rules. Thus, we utilize the concept of a reduct in a new way. Association rules are generated from the original decision table. Each rule is considered as a condition attribute in the newly constructed decision table. The decision attributes are the original decision attributes. Therefore, a reduct of such a decision table represents the essential attributes, which are the most important rules that fully describe the decision. We call these rules *reduct rules*. The reduct rules contained by a reduct are therefore important, and all the other rules are not as important or as representative.

A rule importance measure is introduced to evaluate association rules based on rough sets theory. Multiple reducts are generated by the genetic algorithm from ROSETTA rough sets package [37]. For each reduct, the original data set is reconstructed by using the attributes in the reduct as condition attributes. Multiple rule sets are generated for each new constructed data set by the apriori association rule algorithm. The rules that occur more frequently among the multiple rule sets are considered more important. *Rule importance* measure is defined as follows:

Definition 1 If a rule is generated more frequently across different rule sets, we say this rule is more important than rules generated less frequently across those same rule sets.

Rule importance measure is defined as follows:

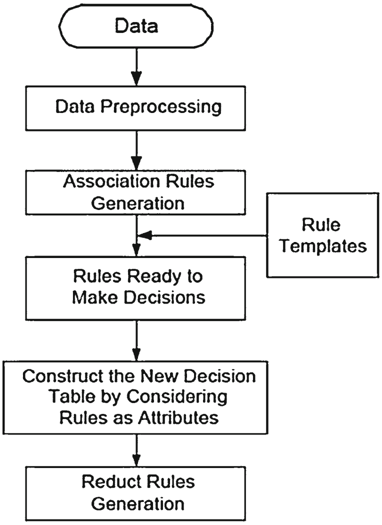
Definition 2 Rule importance measure = Number of times a rule appears in all the generated rules from the reduct sets/Number of reduct sets

This measure is shown to be an effective way of automatically evaluating how important is a rule [37]. The following example shows how to compute the rule importance measure. We use the iris data set from the UCI repository as an example. There are $n = 4$ reducts available for rule generations. The rule sets generated based on the four reducts are shown in Table 2.

Rule sepalLength4.4 → setosa is generated across three rule sets; therefore, the rule importance (RIM) = 3/4 = 75%. For rules sepalWidth 2.9 → versicolor, petalLength1.9 → setosa, petalWidth1.1 → versicolor, they are all generated from 3 of the 4 rule sets; therefore, their rule importance is 75%. The rule importance for the rest rules can be found in [37].

A general problem with rule generation is how to automatically extract important rules from the large number of generated rules. Based on rough sets theory, we propose a new approach. When a reduct is given, rules extracted based on this reduct are representative of the original decision table. These representative rules are therefore considered more important than the rules generated without using the reduct since the reduct contains the most

Fig. 4 Experiment procedure


Data
Data Preprocessing
Association Rules Generation
Rule Templates
Rules Ready to Make Decisions
Construct the New Decision Table by Considering Rules as Attributes
Reduct Rules Generation

representative and important condition attributes of a decision table (the core). Based on this intuition, each of the individual rules among the generated rules sets can be considered as a condition attribute in a decision table. The reduct extracted for such decision tables would contain representative and important attributes, which are the rules. Since the generation of reduct is automatic, we use this approach to discover important rules from the generated rules set automatically.

Figure 4 illustrates our experimental procedure. We consider each data set as a transaction set. During the data preprocessing step, the inconsistent data instances and the data instances containing missing attribute values are processed. The core algorithm ("Appendix C") requires a consistent data set. Therefore in our experiments, the inconsistent data instances are considered as noise and are removed during the data preprocessing stage. Inconsistency exists in a decision table when two or more data instances contain the same condition attribute values but different decision attribute values. We first sort the entire data set according to the condition attributes, excluding the decision attributes. Then, we select data instances that contain the same condition attributes values but different decision attributes values. These data instances are inconsistent, and they are removed during this stage. Discretization, such as equal frequency binning or an entropy-based algorithm is also applied during this stage if necessary. Core attributes are generated at the end of the data preprocessing stage. Apriori association rule algorithm is then applied to generate association rules for each data set. Since our interest is to make decisions, we use the rule templates [36] to generate only rules with decision attributes on the consequent part and to remove subsumed rules. The new decision table is constructed by using these association rules as condition attributes. Note that there may be an inconsistency existing in the new decision table; therefore, the data instances must be removed. We use Johnson's reduct generation algorithm in ROSETTA on the new decision table to generate reduct rules. Other reduct generation approaches may also be applied.

We apply this experimental procedure to a geriatric care data, an actual data set from Dalhousie University Faculty of Medicine, to determine the survival status of a patient. It is used to illustrate that the methods we devised can scale to larger data sets.

Data description. A sanitized geriatric care data set is used as our test data set. This data set contains 8,547 patient records with 44 symptoms and their survival status. This data set is

Table 3 Sample geriatric care data from Dalhousie University Medical School

Edulevel	Eyesight	…	Trouble	Livealone	Cough	Hbp	Heart	…	Studyage	Sex	Livedead
0.6364	0.25	…	0.00	0.00	0.00	0.00	0.00	…	73.00	1.00	0
0.7273	0.50	…	0.50	0.00	0.00	0.00	0.00	…	70.00	2.00	0
0.9091	0.25	…	0.00	0.00	0.00	1.00	1.00	…	76.00	1.00	0
0.5455	0.25	…	0.00	1.00	1.00	0.00	0.00	…	81.00	2.00	0
0.4545	0.25	…	0.00	1.00	0.00	1.00	0.00	…	76.00	2.00	1
0.2727	0.00	…	0.50	1.00	0.00	1.00	0.00	…	76.00	2.00	0
0.8182	0.00	…	0.00	0.00	0.00	1.00	0.00	…	76.00	2.00	1

an actual data set from Dalhousie University Medicine School used to determine the survival status of a patient giving all the symptoms he or she shows. About 3,458 patients are men and 5,089 are women. We use survival status as the decision attribute, and the 44 symptoms of a patient as condition attributes, which consists of different health conditions, level of daily activities, environment stress, age and gender. Sample attributes include education level, the eyesight, hearing, gender, whether the patient can walk, whether the patient can do housework, whether the patient live alone, whether the patient has high blood pressure, heart disease, the age of the patient at investigation and so on. There are no missing values in this data set. Table 3 shows sample data from the geriatric care data.

We first preprocessed the data to remove inconsistent data records from the decision table. Then, multiple reduct sets from Rosetta rough set software as well as the core attributes are obtained to compute the RIM. As an example, one of the 86 generated multiple reducts is, {*edulevel, eyesight, hearing, shopping, housewk, health, trouble, livealone, cough, sneeze, hbp, heart, arthriti, eyetrouble, artroub, dental, chest, kidney, diabetes, feet, nerves, skin, studyage, sex*}. The reducts contain important and representative attributes of a given decision table. The core attributes for the geriatric care data set are, *eartroub, livealone, heart, hbp, eyetroub, hearing, sex, health, edulevel, chest, housewk, diabetes, dental, studyage*. The core attributes are the most important attributes in a given data set, and are used to validate the important discovered rules as medical evidence.

We first check for inconsistency in this data set, and 12 inconsistent data records are removed. Then, 86 reducts are generated by the genetic algorithm in ROSETTA. The apriori algorithm is then used to generate 86 rule sets with support = 30%, confidence = 80%. Rule templates are applied in the rule generation as well, i.e., extracting only rules with decision attribute *livedead* on the consequent part, and removing subsumed rules. For example, in the rule set, a rule such as $SeriousChestProblem \rightarrow Death$ makes removal of the following rule possible because of the subsuming $SeriousChestProblem, TakeMedicineProblem \rightarrow Death$. 218 unique rules are generated over these 86 reducts. These rules as well as their rule importance are shown in Table 4.[14] Among these 218 rules, 87 rules have rule importance of no less than 50%, 8 of which have rule importance of 100%. All the rules with rule importance of 100% contain only core attributes. The core attributes for this data set are *eartrouble, livealone, heart, highbloodpressure, eyetrouble, hearing, sex, health, educationlevel, chest, housework, diabetes, dental,* and *studyage*.

[14] There are 1615 rules generated by apriori algorithm from the original data set with support = 30%, confidence = 80%, after applying the rule template. We can circumvent problems inherent in considering all 1,615 generated rules using the 218 unique rules that are derived from the 86 reducts obtained by ROSETTA's genetic algorithm.

Table 4 Sample rule importance for the geriatric care data set

No.	Selected rules	Rule importance (%)
1	Serious chest problem has a negative effect on the survival status	100
2	Serious hearing problem and diabetes together have a negative effect on the survival status	100
3	Serious ear problem has a negative effect on the survival status	100
4	Serious heart problem has a negative effect on the survival status	100
5	Live alone, having diabetes and having high blood pressure together have a negative effect on the survival status	100
…	…	…
11	Live alone, having diabetes and having nerve problem together have a negative effect on the survival status	95.35
…	…	…
14	Live alone, often cough and having diabetes together have a negative effect on the survival status	93.02
…	…	…
217	Serious hearing problem, and having problem using the phone together have a negative effect on the survival status	1.16
218	Having problem taking medicines and having nerve problem together have a negative effect on the survival status	1.16

Table 5 Reduct rules for the geriatric care data set using 218 rules as condition attributes

No.	Reduct rules	Rule importance (%)
0	SeriousHeartProblem→Death	100.0
1	SeriousChestProblem→Death	100.0
3	SeriousEarProblem→Death	100.0
5	Sex Female→Death	100.0
19	Livealone, OftenSneeze, DentalProblems, HavingDiabetes→Death	82.56
173	ProblemHandleYourOwnMoney→Death	27.91

A new decision table is constructed by using the 218 rules as condition attributes, and the original decision attribute as the decision attribute. Note that after reconstructing the decision table, we must check for inconsistency again before generating reduct rules for this table. After removing the inconsistent data records, there are 5,709 records left in the new decision table. The core rule set is empty. We use Johnson's reduct generation algorithm on this table and the reduct rule set is $\{Rule0, Rule1, Rule3, Rule5, Rule19, Rule173\}$ as shown in Table 5.

Discussion The set of rules as medical evidence obtained from the RIM is evaluated and shown to capture more important knowledge from the medical data. Instead of searching for evidence manually over thousands of generated rules, the RIM provides a promising mechanism for extracting significant medical evidence. In medical diagnosis, when the focus of knowledge discovery is on the important symptoms, the RIM can help facilitate evaluating important knowledge. The RIM is an automatic and objective approach to extract and rank important knowledge. By considering as many reduct sets as possible, we try to cover all representative subsets of the original data set. This measure can also be used jointly with other rule interestingness measures [25] to facilitate the evaluation of the association rules.

Finding evidence from association rule algorithms provides yet additional evidence from medical health data for the overall decision support system charged with finding best evidence for evidence-based best practice recommendations for healthcare. The next section provides an example of validating evidence from multiple sources using multiply section Bayesian networks.

6 Examining selected techniques more closely: validating evidence using multiply sectioned Bayesian networks (MSBNs)

6.1 Overview of MSBNs

MSBNs provide a coherent framework for probabilistic reasoning in distributed interpretation systems with uncertainties, which have been applied in many areas such as medical diagnosis, equipment monitoring and diagnosis and distributed network intrusion detection.

An MSBN is composed of a set of Bayesian subnetworks. A Bayesian network (BN) is a triplet (U, G, P), where U is a set of domain variables, G is a connected DAG, and there is a one-to-one correspondence between nodes in G and variables in V. P is a set of probability distributions: $P = \{P(v|\pi(v))|v \in V\}$, and where $\pi(v)$ denotes the set of parents of v in G. G encodes conditional independencies among variables in U.

In an MSBN, a set of $n > 1$ agents $A_0, A_1, \ldots, A_{n-1}$ populates a total universe U of variables. Each A_i has knowledge over a subdomain $U_i \subset \mathrm{U}$ encoded as a Bayesian subnet (U_i, G_i, P_i). The collection $\{G_0, G_1, \ldots, G_{n-1}\}$ of local DAGs encodes agents' knowledge of domain dependencies. Local DAGs of an MSBN should overlap and be organized into a *hypertree*.[15] For a hyperlink to separate its two directed branches, it has to be a *d-sepset.*[16]

The overall structure of an MSBN is a hypertree MSDAG.[17]

That is, an MSBN is a set of Bayesian subnets whose structures are organized into a hypertree MSDAG. Figure 5 from [7] illustrates the structure of a small MSBN and its hypertree organization, where in (a) each dotted box represents a Bayesian subnet, and in (b) each rectangular box with rounded corner denotes a hypernode. The interface nodes are highlighted with double circles.

In an MSBN, only the nodes in an agent interface are public to the corresponding agents. Otherwise, nodes are private and are known to the corresponding agents only. Agents' privacy forms the constraint of many operations in an MSBN, e.g., triangulation, interface verification and communication. In an MSBN, agents update their beliefs with their local evidence

[15] *Definition of hypertree* (*Xiang* [59]). Let $G = (V, E)$ be a connected graph sectioned into connected subgraphs $\{G_i = (V_i, E_i)\}$. Let these subgraphs be organized into a connected tree Ψ where each node, called a *hypernode*, is labeled by G_i and each link between G_i and G_j , called a *hyperlink*, is labeled by the interface $V_i \bigcap V_j$ such that for each pair of nodes G_l and G_m, $V_l \bigcap V_m$ is contained in each subgraph on the path between G_l and G_m. The tree Ψ is called a *hypertree* over G.

[16] *d-sepset* (*Xiang* [59]). Let G be a directed graph such that a hypertree over G exists. A node x contained in more than one subgraph with its parents $\pi(x)$ in G is a *d-sepnode* if there exists a subgraph that contains $\pi(x)$. An interface I is a *d-sepset* if every $x \in I$ is a *d-sepnode*.

[17] *hypertree MSDAG* (*Xiang* [59]). A hypertree MSDAG $G = \cup_i G_i$, where each $G_i = (V_i, E_i)$ is a DAG, is a connected DAG such that there exists a hypertree over G and each hyperlink is a d-sepset.

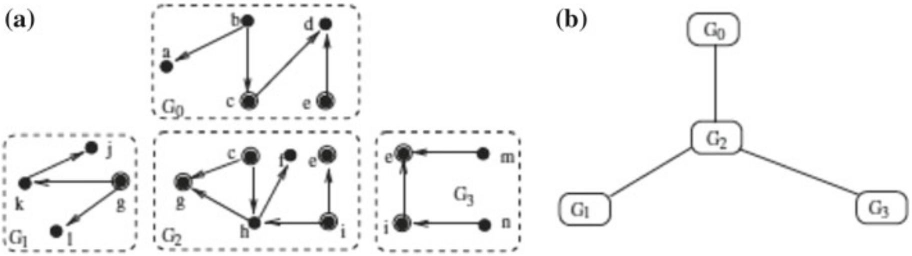


Fig. 5 **a** The structure of an MSBN and **b** the hypertree of the MSBN

and global information from other agents, and then answer questions or take actions based on their beliefs.

In addition to [59], readers can obtain more extended information on MSBNs from [5–7,9].

6.2 An example of validating evidence for multiple diseases (MSBNs)

In this section, we use an example to show how to validate evidence with MSBNs for best practice recommendation in health care. In particular, we illustrate how to validate evidence for multiple diseases diagnosis using MSBNs in an object-oriented way.

In the example, we use an MSBN to model the diagnostic knowledge of 3 groups of cardiologic diseases: diseases of aorta, pericardial diseases and cardiomyopathies. The diagnostic knowledge is originally presented in a set of tables which provides probability of a disease given each set of observed symptoms and lab results. The knowledge has been validated by medical experts. However, we could not use such diagnostic knowledge directly due to the exponential size of tables. Instead, we calculate causal knowledge from the diagnostic knowledge to construct MSBNs. Causal knowledge tells us the probability of each symptom or lab result given a disease which can be stored in very small tables. It is shown [8] the BNs made from such causal knowledge is very efficient and the diagnosis from these BNs is very consistent with diagnostic knowledge tables. In this example, we use causal knowledge obtained from diagnostic knowledge tables to construct 3 BNs and then merge them into an MSBN.

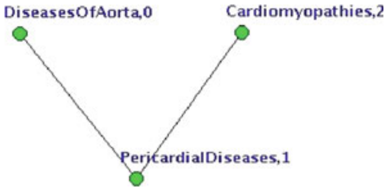
In an MSBN, all subnets are connected into a tree structure via the shared variables. Each subnet is a directed acyclic graph. So, an MSBN can be described in two levels: subnet tree organization, and directed acyclic subnets.

Figure 6 shows the MSBN of 3 BN models, each of which is denoted by a green node. Beside each green node is its label followed by a unique index number (for internal use only). Each BN model represents the medical knowledge of a group of cardiologic diseases. The 3 groups of diseases represented here are diseases of aorta (node 0), pericardial diseases (node 1) and cardiomyopathies (node 2). Next, we look at each BN model one by one in detail.

The Bayesian model representing the knowledge of the group of diseases of aorta is as shown by Fig. 7, where node 0 represents the disease of aortic dissection, node 11 the disease of thoracic aortic aneurysm and node 16 the disease of abdominal aortic aneurysm. Different from Fig. 6, each green node here denotes a random variable representing a disease or a lab test.

The Bayesian model representing the knowledge of the pericardial diseases is as shown by Fig. 8, where node 8 represents acute pericarditis and node 0 constrictive pericarditis.

Fig. 6 An MSBN of 3 BN medical models



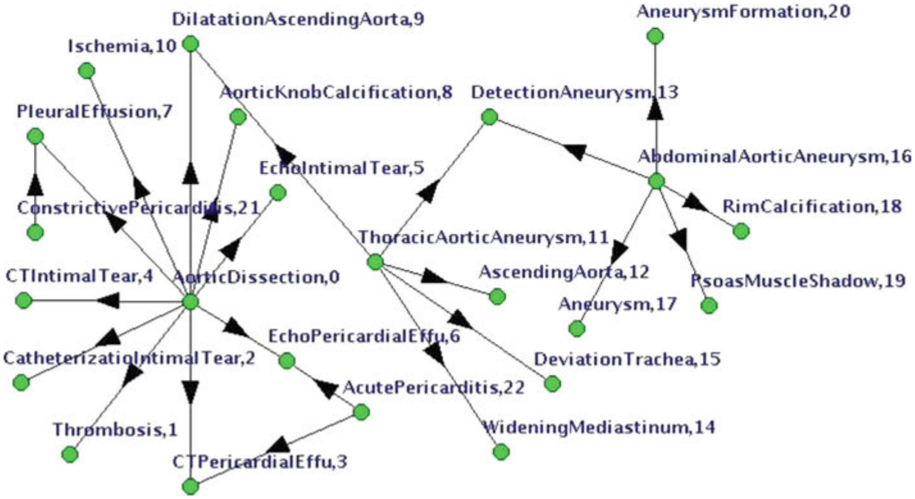


Fig. 7 The BN model of the diseases of aorta

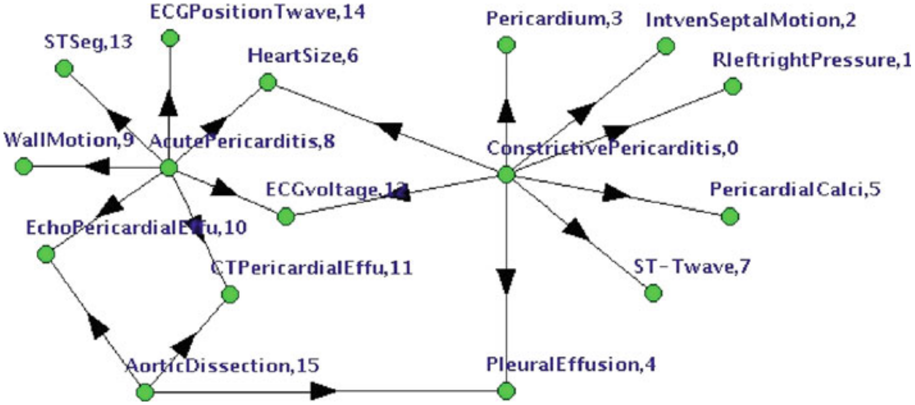


Fig. 8 The BN model of the pericardial diseases

The Bayesian subnet representing the knowledge of cardiomyopathies is shown in Fig. 9, where node 1 represents dilated cardiomyopathy, node 11 restrictive cardiomyopathy and node 16 hypertrophic cardiomyopathy.

Before illustrating how evidence is validated, we first have a discussion on how the probabilities are calculated in MSBNs. In this example, we show a set of subnets that encodes

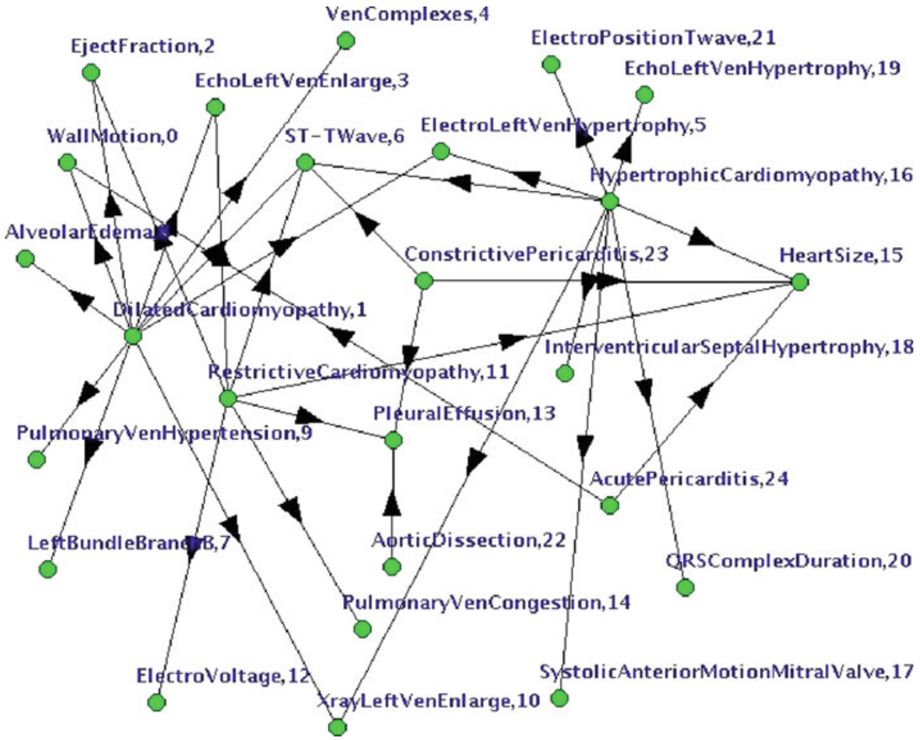


Fig. 9 The BN model of cardiomyopathies

qualitative dependencies among variables in the problem domain. We do not show the set of probability distribution tables used to parameterize the strength of dependencies since the numerous small tables are not so relevant in understanding the example but distract readers from its focus. However, it is absolutely helpful to give a simple (intuitive) discussion on how the probability is calculated and propagated in the MSBN.

The probability distribution tables determine the strength of dependencies in MSBNs. For the MSBN we illustrate above, the probability distribution tables are specified such that the probability of a positive symptom (e.g., running nose) is quite high given the disease is true (e.g., flu) and quite low given the disease is false. So for one more positive symptom confirmed, the disease becomes more positive (higher probability). When the belief on one variable changes, the belief on all other variables need to be updated since all variables in the domain are related, directly or indirectly, strongly or weakly. The belief change at one variable would have stronger impact on the belief of another variable if they depend on each other directly and strongly. The belief is updated based on Bayes' rule.

Information not only flows from one node to another in one subnet but also from one subnet to another subnet through their interfaces (shared variables). So, the impact of new knowledge would arrive at any subnets after communication.

Figure 10 shows the belief on diseases of aorta before any lab tests are available. The histograms denote the distributions of possible test outcomes or disease occurrences. At this time, possible outcomes of lab tests are uniformly distributed since we do not have any knowledge about them. The probabilities of disease occurrences are pretty low.

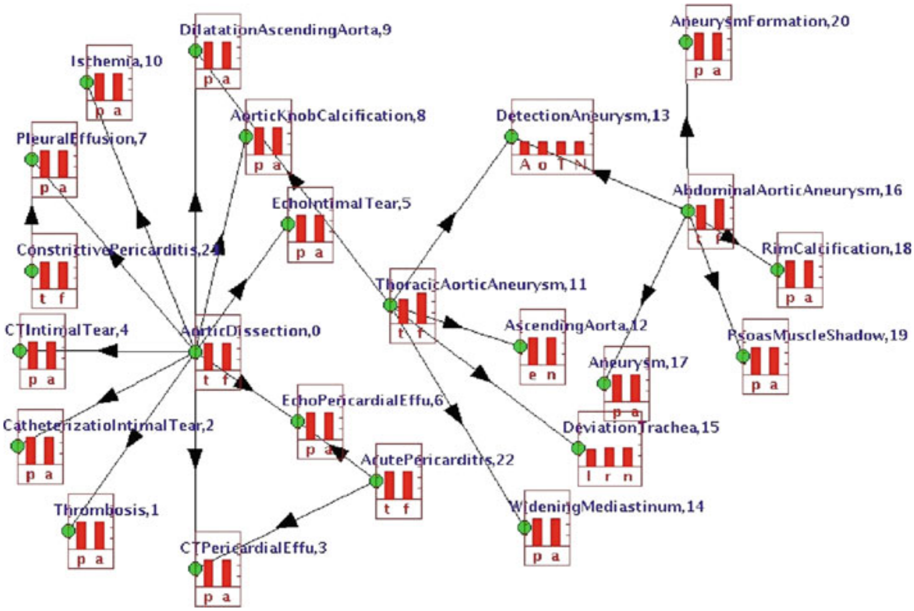


Fig. 10 Prior belief of the diseases of aorta

Assume we get some test results next as shown in Fig. 11, where each yellow node indicates a confirmed test result. The probability of aortic dissection (node 0) becomes pretty high since all relevant tests show positive. The outcome for the detection of aneurysm (node 13) makes thoracic aortic aneurysm (node 11) less possible but abdominal aortic aneurysm (node 16) more possible since this result is negative for the former but positive for the latter. The probability of aneurysm formation (node 20) becomes higher due to the rise of the occurrence probability of abdominal aortic aneurysm. From Fig. 11, we can see how the information (influence) flows from one variable to another, and how variables interact.

Through communication, information further flows from the subnet of the diseases of aorta to the subnet of the pericaridal diseases. As shown in Fig. 12, yellow nodes (10, 11) indicate the test results are confirmed from other subnets. The probability of acute pericarditis (node 8) becomes higher due to the two positive test results.

Therefore, by MSBNs that are constructed based on medical knowledge, we may interpret and validate medical evidence acquired from different sources.

7 Concluding remarks

Our primary long-term goal of this research is to develop a decision support system for evidence-based management, quality control and best practice recommendations in the area of medical prescriptions, broadly conceived. Specifically, our research should improve accessibility, management and manipulation of information in a networked environment by: (1) building an application prototype for adaptive information extraction (IE), text and data mining from (online) documents to find evidence on which to base best practice recommendations; and (2) employing multiply sectioned Bayesian networks (MSBNs) to infer

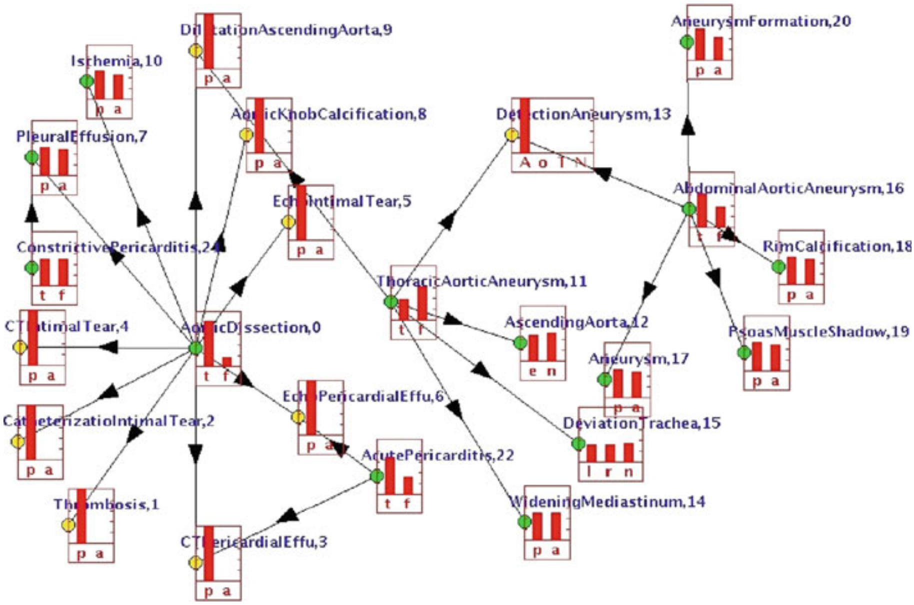


Fig. 11 Diagnosis at the diseases of aorta model

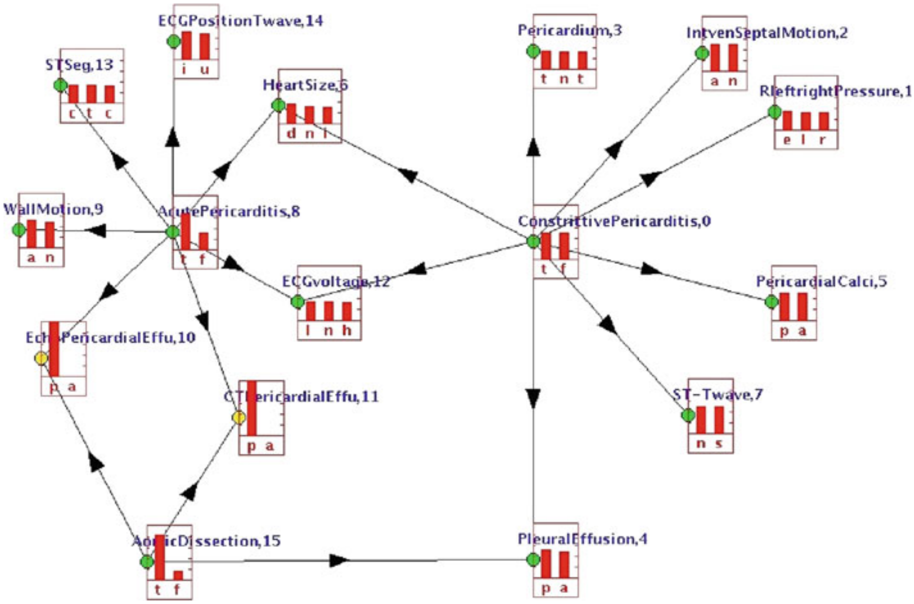


Fig. 12 Information flow from one model to another

a probabilistic interpretation to validate evidence for recommendations (MSBNs provide a framework for probabilistic inference).

We presented a general overview of techniques and methods including work on facts, factoids and frames. Terminology extraction and work beyond factoids (discovering interesting patterns from health care databases, adaptive medical event sequence clustering with Markov models and direct and indirect association mining), along with validating evidence for recommendation (using MSBNs) were discussed.

We then examined selected techniques and methods more closely in greater detail including naïve Bayes IE**,** demonstrating a method to discover evidence from individual encounters using a classification algorithm, HMMs for segment-based text retrieval, and discussing how to validate evidence for multiple diseases using multiply sectioned Bayesian networks (MSBNs).

In the Bayes IE technique, we observed problems in previous work on applying naive Bayes for information extraction in their formulation of information extraction as a maximum a posteriori (MAP) learning problem, ignoring certain aspects inappropriately. In view of this formulation problem, we illustrated how to formulate an information extraction task to a Bayesian learning problem. Then based on the formulation, we showed that various naive Bayes models, from the simplest one to more complicated models, can be induced by making different assumptions on probability estimation with regard to its dependence to contextual information.

In the MSBN discussion, the MSBN can be described in two levels: subnet tree organization, and directed acyclic subnets. When the belief on one variable changes, the belief on all other variables need to be updated since all variables in the domain are related, directly or indirectly, strongly or weakly. The belief change at one variable would have stronger impact on the belief of another variable if they depend on each other directly and strongly. Information can flow from one subnet to another subnet because they share some variables.

Obviously, much work remains to be done. What we have reported is the initial design and some early results of a large complex project that involves five PI's, three post-docs and half a dozen graduate students. We receive wonderful support from our granting council, NSERC, Canadian industry and Canadian government agencies at both Federal and Provincial levels. We look forward to produce a new research over the next year in hopes of reporting a substantive system development at that time.

Appendix A: Experimental results of the four IE learners and performance comparison to related work

For these experiments, we use the "seminar announcements" domain data set which contains 485 annotated seminar announcement texts, with the four slots (i.e., location, speaker, stime and etime) tagged in XML format. Tenfold cross validation is performed on these annotated data to evaluate extraction performance of these naive Bayes IE learners on the seminar announcement domain. We also report the effect of various elements in a naive Bayes IE system such as smoothing methods, threshold and context size setting, context data likelihood estimate in this section via experiments, as well as system performance comparison to the related work.

Table 6 shows the precision and recall numbers of all these four IE learners (i.e., Bayes Basic, Bayes Context, Bayes ContextPro and Bayes Position) combined with three different smoothing settings, i.e., no smoothing (None), our modified Laplace smoothing (Lap) and the simple good-turning smoothing (SGT). Both the Laplace smoothing and the SGT smoothing applied here used unseen species estimate and the probability assignment among unseen events as introduced in [23]. Exact matching criteria is used for extraction correctness evaluation, in other words, an issued extraction is considered as a correct extraction for that document only when the text segment extracted exactly matches the filler in the labeled data

Table 6 Precision/recall results on seminar announcements

Learner	Location	Speaker	stime	etime
Bayes Basic None	0.6063/0.6335	0.1844/0.1892	0.5713/0.5713	0.3442/0.7327
Bayes Basic Lap	0.4601/0.4806	0.3792/0.4415	0.5238/0.5238	0.2864/0.6104
Bayes Basic SGT	0.6187/0.6465	0.4412/0.5045	0.5713/0.5713	0.3442/0.7327
Bayes Context None	0.0152/0.0149	0.0170/0.0173	0.0496/0.0496	0.1774/0.3078
Bayes Context Lap	0.0167/0.0170	0.0243/0.0263	0.0476/0.0476	0.1673/0.3127
Bayes Context SGT	0.0167/0.0171	0.0243/0.0261	0.0558/0.0558	0.1689/0.3161
Bayes Context Pro None	0.4963/0.3110	0.6551/0.1364	0.3390/0.3386	0.5650/0.6653
Bayes Context Pro Lap	0.6264/0.5277	0.6667/0.4520	0.3571/0.3571	0.5823/0.8862
Bayes Context Pro SGT	0.6052/0.6029	0.5350/0.5293	0.3531/0.3531	0.4765/0.8939
Bayes Position None	0.8341/0.2872	0.9667/0.1242	0.9735/0.7754	0.9775/0.6305
Bayes Position Lap	0.8113/0.5113	0.8013/0.3831	0.9856/0.9835	0.9390/0.9394
Bayes Position SGT	0.7560/0.6271	0.7103/0.5280	0.9856/0.9856	0.9387/0.9481

Table 7 $F1$ results on seminar announcements

Learner	Location	Speaker	stime	etime
Bayes Basic None	0.6196	0.1868	0.5713	0.4684
Bayes Basic Lap	0.4701	0.4080	0.5238	0.3899
Bayes Basic SGT	0.6323	0.4707	0.5713	0.4684
Bayes Context None	0.0150	0.0171	0.0496	0.2251
Bayes Context Lap	0.0168	0.0253	0.0476	0.2180
Bayes Context SGT	0.0169	0.0252	0.0558	0.2202
Bayes Context Pro None	0.3824	0.2258	0.3388	0.6111
Bayes Context Pro Lap	0.5728	0.5387	0.3571	0.7028
Bayes Context Pro SGT	0.6040	0.5321	0.3531	0.6216
Bayes Position None	0.4273	0.2201	0.8632	0.7666
Bayes Position Lap	0.6273	0.5184	0.9845	0.9392
Bayes Position SGT	0.6855	0.6057	0.9856	0.9434

(i.e., the key filler). As to the parameter setting in the naive Bayes IE systems, we enforce the use of threshold on the extractions to be issued by the system, and the threshold is learned from a second run over the training set during the training phase. Context window sizes for all slots are set to the default setting, i.e., 5 tokens on either side of fillers, in this experiment for simplicity.

Table 7 shows $F1$ numbers of all four IE learners combined with three different smoothing settings.

Among the four naive Bayes IE learners we studied, Bayes Context is the worst learner in detecting exactly matched extractions. This result is due to the fact that in Bayes Context model, tokens within the context are also considered as filler indicators, while position information of their appearance is neglected in the modeling. Ignorance of token ordering would cause a major problem in determining boundaries between the fillers and the tokens

Table 8 $F1$ results comparison with no smoothing on seminar announcement

Learner	Location	Speaker	stime	etime
Bayes Basic None	0.6196	0.1868	0.5713	0.4684
Bayes Context None	0.0150	0.0171	0.0496	0.2251
Bayes ContextPro None	0.3824	0.2258	0.3388	0.6111
Bayes Position None	0.4273	0.2201	0.3388	0.7666

in the context appearing right before and after the fillers. Our context size is relatively large (5 tokens on both side) in this experiment which also make this problem worse.

Combined with the power of smoothing, Bayes Position is obviously the best learner in this experiment. Compared with Bayes ContextPro, Bayes Basic performs better in "location" and "stime" slots, while Bayes ContextPro performs better in "speaker" and "etime". However, it is noted that we did not allow the Bayes ContextPro system learn its optimal context size in this experiment, which could keep this learner from performing at its optimal status.

When we compare all these learners' performance for their no smoothing case (Table 8), it is observed that Bayes Position with no smoothing achieves the best performance only for two of the four slots tested, i.e., "stime" and "etime". Bayes ContextPro has slightly better overall performance on "speaker". Bayes Basic performs significantly better on "location" slot than the other three more advanced learners in which certain contextual information is considered. The reason of this observation might be due to the characteristics of the *location* slot itself as well as the domain data we have at hand, in which all the labeled seminar announcements are from the posts in the same department. In this case, the set of the text location fillers appearing in these seminar announcements is more like a closed-class of proper nouns, and most of these fillers might be observed in the training set. The filler tokens are strong indicators for such kinds of slot, and adding more contextual information in the modeling would not help significantly on the extraction.

The reason for the most advanced model Bayes Position not being able to give the best performance results for all slots when no smoothing is used, is due to the training problem of data sparseness. Compared to other naive Bayes IE models, many more probabilities need to be estimated in Bayes Position from the same set of annotated examples. The training set is not sufficient for this advanced model to obtain stable probability estimates without smoothing, because there are a larger number of unseen events for those term distributions to be estimated by term counting from the training texts.

Effect of smoothing

The general effect of smoothing in probability estimate can also be observed from Table 7. In general, smoothing helps the learner to get more probability estimation, especially for more advanced learners such as Bayes ContextPro and Bayes Position. In particular, as we can seen from Table 6 in precision/recall numbers that the improvement caused by using the Laplace smoothing or the SGT smoothing, mainly comes from the increase of *recall* compared to the results from no smoothing setting. Comparisons between the two smoothing methods in which we have experimented show that, SGT is obviously a better smoothing method overall. Although the Laplace smoothing can boost extraction performance in many cases, compared to the no smoothing case, its effect is unstable since the overall performance has

Table 9 $F1$ comparison with related system on seminar announcements

Learner	Location	Speaker	stime	etime
Bayes ContextPro $F1$	0.6538	0.4812	0.9579	0.8481
Bayes Position $F1$	0.6346	0.5067	0.9851	0.9575
BayesIDF $F1$ (peak $F1$ from [19])	0.613	0.297	0.982	0.923

been deteriorated in some other cases by applying the Laplace smoothing (such as for Bayes Basic Lap performances on all slots except “speaker”). On the other hand, SGT shows more consistent and positive smoothing effect, in which the performance with the use of SGT is almost always better than none-smoothing cases (with the only exception at Bayes Context SGT for “etime” being slightly worse than Bayes-Context None). In most cases, the SGT smoothing can lead to larger performance increase than the Laplace smoothing.

We compare our naive Bayes IE systems to the related work reported in [19]. We run our Bayes ContextPro and Bayes Position IE systems using the same evaluation setting on the seminar announcement collection, which randomly divides the whole labeled seminars into two equally sized training set and testing set and runs five times to get the average performance numbers. The performance measures used in [19] are also precision and recall, but they reported peak $F1$ (i.e., the maximum $F1$ score achieved by a learner at any point on its precision/recall curve) scores for comparing system overall performance. We cited the peak $F1$ scores of their best version of Bayes learner named as BayesIDF, in Table 9 to be compared with the $F1$ numbers from our two naive Bayes learners.

Table 9 clearly shows that our naive Bayes Position learner performs consistently better than their BayesIDF system on all the slots in this domain, with significantly improvement on “speaker”. We regard the major performance improvement is gained from our correct naive Bayes modeling for an IE problem, as well as the properly designed smoothing method.

Appendix B: Experimental results on document HMM IE

We tested our two document HMM IE systems Doc HMM Basic and Doc HMM Context on the seminar announcements, using tenfold cross-validation evaluation as we evaluated our naive Bayes IE systems. In our experiments, the structure parameters for HMM Context model are set to system default values, i.e., 4 for both pre-context and post-context size and 4 for the number of parallel filler paths. Table 10 shows $F1$ scores of both Document HMM IE systems. The performance numbers from other HMM IE systems [19] are also listed in Table 10 for comparison, where HMM None is their HMM IE system that uses absolute discounting but with no “shrinkage”, and HMM Global is the representative version of their HMM IE system with “shrinkage”.

By using the same structure parameters for HMM Context as in [20], our Doc HMM Context performs consistently better on all slots than their HMM IE system using absolute discounting. Even compared to their much more complex version of HMM IE with “shrinkage”, our Doc HMM Context achieves comparable results on “location”, “speaker” and “stime”, but obtained significantly better performance on the “etime” slot. It is noted that our smoothing method is much simpler to apply, which does not require any extra effort such as specifying shrinkage topology or any extra labeled data for a “held-out” set.

Table 10 $F1$ of Document HMM IE systems on seminar announcements

Learner	Location	Speaker	stime	etime
Doc HMM Basic	0.6357	0.1415	0.6071	0.5572
Doc HMM Context	0.8220	0.7135	1.0000	0.9488
HMM None	0.735	0.513	0.991	0.814
HMM Global	0.839	0.711	0.991	0.595

Appendix C: Rough sets theory: reduct calculation

An information system can be represented by a decision table, with condition attributes describing the conditions, and decision attributes describing the decisions when certain conditions are satisfied. A decision table can be defined as $T = (U, C, D)$, where U is the set of objects in the table, C is the set of the condition attributes, and D is the set of the decision attributes. A reduct is the essential part that can sufficiently represent all original concepts.

In the original definition for a reduct, U is the set of objects we in which are interested, where $U \neq \varphi$. Let R be an equivalence relation over U, then the family of all equivalence classes of R is represented by U/R. $[x]_R$ means an equivalence class of R containing an element $x \in U$. Suppose $P \subseteq C$, and $P \neq \varphi$, IND(P) is an indiscernible relationship induced by P and an equivalence relation over U. For any $x \in U$, the equivalence class of x of the relation IND(P) is denoted as $[x]_P$. X is a subset of U, and R is an equivalence relation. The lower approximation of X and the upper approximation of X are defined as

$$\underline{R}X = *\{x \in U |\, [x]_R \subseteq X\} \tag{1}$$
$$\overline{R}X = *\{x \in U |\, [x]_R \cap X\_ = \varphi\} \tag{2}$$

Let $P \subseteq C$ and $S \subseteq R$. We say, S is dispensable in P, if IND(P) = IND($P - \{S\}$) and S is indispensable otherwise. We say P is independent if each $S \subseteq P$ is indispensable in P. Q is a *reduct* of P if Q is independent, $Q \subseteq P$, and IND(Q) = IND(P). An indiscernible equivalence relation over a knowledge base can have many reducts.

A *reduct* of a decision table is a set of condition attributes that are sufficient to define the decision attributes. A reduct does not contain redundant attributes. It is often used in the attribute selection process. There may exist more than one reduct for each decision table. Finding all the reduct sets for a data set is NP-hard [33]. Approximation algorithms are used to obtain reduct sets [11], The intersection of all the possible reducts is called the *core*. The core is contained in all the reduct sets, and it is the essential of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

All reducts contain the core. The core represents the most important information of the original data set.

Since it is infeasible to obtain the core attributes by intersecting all the possible reducts, other approaches are proposed to generate the core attributes. Hu et al. introduced a core generation algorithm based on rough sets theory and efficient database operations, without generating reducts [26]. We use Hu's algorithm to obtain core attributes and to examine the effect of core attributes on the generated rules.

The algorithm is shown below, where C is the set of condition attributes, and D is the set of decision attributes. *Card* denotes the database count operation, and Π denotes the projection operation in databases.

Hu's algorithm is developed to consider the effect of each condition attribute on the decision attribute. The intuition is that, if the core attribute is removed from the decision table, the rest of the attributes will cause decisions. Theoretical proof of this algorithm is provided in [26]. The algorithm takes advantage of efficient database operations such as count and projection. Since the attributes of the core are contained in any reduct set for a data set, this algorithm also provides an evaluation to justify the correctness of the reduct sets.

Hu et al. [26] proposed a new rough set model based on database operations such as cardinality and projection. By combining the relational algebra with traditional rough sets theory, the approach is designed to increase the efficiency of the core and reduct computation based on database operations. The reduct is defined to be a subset $REDU (\subseteq C)$ of condition attributes with respect to the decision attribute D, where $REDU$ is a minimum subset of attributes that has the same classification power as the entire condition attributes. Let $K(REDU, D)$ be the proportion of the data instances in the decision table that can be classified. K is also defined to be the degree of dependency and is the stopping criteria for the algorithm, as shown in (3).

$$K(REDU, D) = \frac{Card(\Pi(REDU + D))}{Card(\Pi(C + D))} \tag{3}$$

A measure of *merit value* is defined to evaluate the effect of each condition attribute on the decision attribute D. For a condition attribute $C_i \in C$, the merit of C_i can be calculated by

$$Merit(C_i, C, D) = \frac{Card(\Pi(C - \{Ci\} + D))}{Card(\Pi(C + D))} \tag{4}$$

```
Algorithm Hu: CORE GENERATING ALGORITHM
        input: Decision table $T(C, D)$, $C$ is the condition attributes set; $D$ is the
        decision attribute set.
        output: $Core$, Core attributes set.
$Core \leftarrow \varphi$;
for each condition attribute $A \in C$ do
    if $Card(\Pi(C - \{A\} + D)) \neq Card(\Pi(C - \{A\}))$ then
        $Core = Core + \{A\}$;
    end
end
return $Core$;
```

During the reduct generation, the condition attribute with the highest merit value at the moment is included in the reduct. In case multiple highest merit values exist, the condition attribute with the least combination with other attributes in the current reduct is selected. The algorithm iterates until the minimum set of attributes which is as representative as the entire set of condition attributes is obtained. The reduct generation algorithm, shown below, is designed to guarantee that the generated reduct will have the minimum number of attributes.

```
Algorithm HU-R: HU'S REDUCT GENERATING ALGORITHM
        input: Decision table $T(C, D)$, $C$ is the condition attributes set; $D$ is the
        decision attribute set.
        output: $REDU$, reduct of $C$.
2.1  Core Generation Algorithm to generate $Core$;
2.2  $REDU = Core$;
2.3  $AR = C - REDU$;
2.4  for each attribute $Ci$ ( $AR$ do
2.5      $Merit(Ci,C,D) = 1 - Card(\Pi(C\text{-}\{Ci\}+D)) / Card(\Pi(C + D))$
2.6  end
2.7  maximum ($Merit(Cj,C,D)$);
        /*In case there are several attributes with the same merit value, choose the
        attribute which has the least number of combinations with those
        attributes in $REDU$. minimum($Card(\Pi(\{Cj\} + REDU))$) */
2.8  $REDU = REDU + \{Cj\}, AR = AR - \{Cj\}$;
2.9  if $K(REDU, D) = 1$ then return $REDU$;
2.10   else go to Step 2.4
```

The QuickReduct algorithm was first applied in information retrieval systems to reduce the dimensions of the input text data. This algorithm does not combine database operations. We rewrite the QuickReduct algorithm with the combination of database operations as shown below.

```
Algorithm QR: QUICKREDUCT ALGORITHM
        input: Decision table $T(C, D)$, $C$ is the condition attributes set; $D$ is the
        decision attribute set.
        output: $REDU$, reduct of $C$.
3.1  $REDU \leftarrow \varphi$;
3.2  repeat
3.3      $M \leftarrow REDU$;
3.4  for each attribute $x$ ( $(C - REDU)$ do
3.5      if $Card(\Pi(REDU + \{x\} + D)) > Card(\Pi(M + D))$ then $M \leftarrow REDU + \{x\}$;
3.6  end
3.7      $REDU \leftarrow M$;
3.8  until $Card(\_(REDU + D)) = Card(\Pi(C + D))$;
3.9   return $REDU$;
```

References

1. Abou-Assaleh T, Cercone N, Doyle J, Keselj V, Whidden C (2005) DalTREC 2005 QA system jellyfish: mark-and-match approach to question answering. In: Proceedings of the 14th TREC, Gaithersburg, Maryland, USA
2. Abou-Assaleh T, Cercone N, Keselj V (2005) Question-answering with relaxed unification. In: Proceedings of the PACLING'05, Meisei University, Hino Campus, Hino-shi, Tokyo, 191-8506 Japan
3. Abney S (1996) Partial parsing via finite-state cascades. In: Proceedings of the ESSLI 96 robust parsing workshop
4. Agrawal A, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of VLDB
5. An X (2005) Towards dynamic multiagent probabilistic inference: testbeds and methods. Ph.D. thesis, University of Waterloo, Canada
6. An X, Xiang Y, Cercone N (2008) Dynamic multiagent probabilistic inference. Int J Approx Reason 48: 185–213

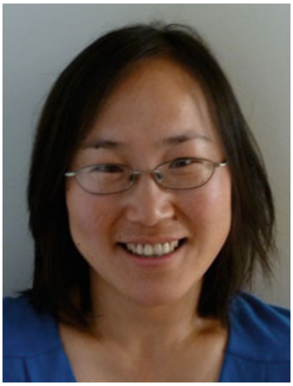
7. An X, Cercone N (2009) Distributed parallel compilation of MSBNs, concurrency and computation: practice and experience. Wiley Interscience, Hoboken
8. An X, Cercone N (2008) Mining causal knowledge from diagnostic knowledge. In: Proceedings of the 4th international conference on advanced data mining and applications (ADMA 2008), pp 556–562
9. An X, Cercone N (2009) Fault-tolerant multiagent exact belief propagation. Comput Intell 25(1):1–30
10. Ananiadou S, Friedmann C, Tsujii J (2004) Named entity recognition in biomedicine. J Biomed Inform 37(6):393–528
11. Bazan J, Nguyen H, Nguyen S, Synak P, Wroblewski J (2000) Rough set algorithms in classification problem. In: Ślezak D (ed) Rough set methods and applications: new developments in knowledge discovery in information systems. Physica-Verlag, Heidelberg, Germany, pp 49–88
12. Bickel S, Brefeld U, Faulstich L, Hakenberg U, Leser J, Plake C, Scheffer T (2004) A support vector machine classifier for gene name recognition. In: BioCreative: EMBO Workshop—a critical assessment of text mining methods in molecular biology, Granada, Spain
13. Biemann C, Quasthoff U (2002) Named entitiy learning and verification: in large corpora. In: CoNLL-2002, the sixth workshop on computational language learning, Morgan Kaufmann, Taipei, Taiwan, San Francisco, pp 8–14
14. Califf M, Mooney RJ (1999) Relational learning of pattern-match rules for information extraction. In: Proceedings of the 16th national conference on AI (AAAI-99), Orlando, FL, pp 328–334
15. Cadez IV, Heckerman D, Meek C, Smyth P, White S (2000) Visualization of navigation patterns on a website using model-based clustering. In: Knowledge discovery and data mining, pp 280–284
16. Ciravegna F (2001) Adaptive information extraction from text by rule induction and generalization. In: Proceedings of the seventeenth international joint conference on artificial intelligence
17. Freitag D (1998) Multistrategy learning for information extraction. In: Proceedings of the fifteenth international conference on machine learning, pp 161–169
18. Freitag D (1997) Using grammatical inference to improve precision in information extraction. In: Proceedings of the ICML'97 workshop on automata induction, grammatical inference, and language acquisition, Nashville, NT, USA
19. Freitag D (1998) Machine learning for information extraction in informal domains. Ph.D. thesis, CMU
20. Freitag D, McCallum A (1999) Information extraction with HMMs and shrinkage. In: Proceedings of the AAAI-99 workshop on machine learning for information extraction
21. Fernandez A, Morales M, Rodriguez C, Salmeron A (2010) A system for relevance analysis of performance indicators in higher education using Bayesian networks. Knowl Inf Syst, published online 1
22. Gale W, Sampson G (1995) Good-turning smoothing without tears. J Quant Linguist 2:217–237
23. Gu Z (2006) Adaptive information extraction from online documents. Ph.D. thesis, U Waterloo, Canada
24. Hishiki, Collier T, Nobata N, Okazaki-Ohta C, Ogata T, Sekimizu N, Steiner T, Park R, Tsujii HS (1998) J. Developing NLP tools for genome informatics: an IE perspective. Genome Info., SERS 9, pp 81–90
25. Huang, X, An, A, Cercone, N, Promhouse, G (2002) Discovery of interesting association rules from livelink web log data. In: Proceedings of the IEEE ICDM, Maebashi, Japan
26. Hu X, Lin T, Han J (2004) A new rough sets model based on database systems. Fundam Inform 59(2–3): 135–152
27. Jalali V, Reza M, Borujerdi M (2010) Information retrieval with concept-based pseudo-relevance feedback in MEDLINE. Knowl Inf Syst, published online 21
28. Jiampojamarn S, Keselj V, Cercone N (2004) Two experiments in biological term annotation using classification methods. In: Proceedings of the first BIOT-04, Colorado Springs, Colorado, USA
29. Kazamay J, Makinoz T, Ohta Y, Tsujiiy J (2002) Tuning support vector machines for biomedical named entity recognition. In: Proceedings of ACL. Workshop on NLP in the biomedical domain (workshop)
30. Keselj V, Cox A (2004) DalTREC 2004: question answering using regular expression rewriting. In: Proceedings of the thirteenth text REtrieval conference (TREC 2004), Gaithersburg, Maryland, USA
31. Keselj V (2002) Modular stochastic HPSGs for question answering. Ph.D. Thesis, University of Waterloo, Waterloo, Canada
32. Klemettinen M, Mannila H, Ronkainen P, Toivonen H, Verkamo AI (1994) Finding interesting rules from large sets of discovered association rules. In: Adam NR, Bhargava BK, Yesha Y (eds) Proceedings of the third international conference on information and knowledge management (CIKMí94). ACM Press, New York, pp 401–407
33. Kryszkiewicz M, Rybinski H (1993) Finding reducts in composed information systems. In: Ziarko W (ed) Proceedings of the international workshop on rough sets, knowledge discovery, pp 261–273
34. Kiritchenko S, Matwin S, Famili F (2004) Hierarchical text categorization as a tool of associating genes with gene ontology codes, ECML/PKDD 2004 workshop on data mining and text mining for bioinformatics, Pisa

35. Li J (2007) Rough set based rule evaluations and their applications. Ph.D. thesis, University of Waterloo, Canada
36. Li J, Cercone N (2010) A method of discovering important rules using rules as attributes. Int J Intell Syst 25:180–206
37. Li J, Cercone N (2005) A rough set based model to rank the importance of association rules. In: RSFDGrC (2), pp 109–118
38. Liu Y, Huang X, An A, Promhouse G (2004) Clustering web surfers with probabilistic models in a real application. In: Web Intelligence 2004, pp 761–765
39. Liu Y, Huang X, An A (2007) Personalized recommendation with adaptive mixture of markov models. J Am Soc Inf Sci Technol 58(12):1851–1870, Wiley
40. McCallum A, Freitag D, Pereira (2000) Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of ICML-2000
41. MPUP. http://www.ccohta.ca/mpup/pdf/faq_e.pdf
42. Manning CD, Schutze H (1999) Foundations of statistical natural language processing, chapter 15: topics in information retrieval, pp 529–573. The MIT Press, Cambridge
43. Mielikäinen T (2005) Summarization techniques for pattern collections in data mining. Ph.D. thesis, University of Helsinki
44. Mooney R, Bunescu R (2005) Mining knowledge from text using information extraction. In: ACM SIGKDD explorations newsletter
45. Ohta T, Tateisi Y, Kim J, Mima H, Tsujii J (2002) The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of human language technology conference
46. Ohta T, Tateisi Y, Kim J, Mima H, Tsujii J (2002) The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of human language technology conference
47. Øhrn A (1999) Discernibility and rough sets in medicine: tools and applications. Ph.D. thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim Norway
48. Pascot D, Bouslama F, Mellouli S (2010) Architecturing large integrated complex information systems: an application to healthcare. Knowl Inf Syst, published online 19
49. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo
50. Rindfleisch TC, Tanabe L, Weinstein JN (2000) EDGAR: extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the Pacific symposium on biocomputing
51. Roche E, Shabes Y (1997) Finite-state language processing. MIT Press, Cambridge
52. Ramirez J, Cook D, Peterson L, Peterson D (2000) An event set approach to sequence discovery in medical data. Intell Data Anal 4(6/2000):513–530, IOS Press
53. Saragawi S, Cohen W (2004) Semi-markov conditional random fields for information extraction. In: Advances in neural information processing systems—NIPS
54. Satti A, Cercone N, Keselj V (2004) Experiments in web page classification for semantic web. WSS'04, With 2004 IEEE/WIC/ACM international conference on web intelligence, Beijing, China
55. Soderland S (1999) Learning information extraction rules for semi-structured and free text. Mach Learn 34: 233–272
56. Smyth P (1996) Clustering sequences with hidden markov models. In: Proceedings of the NIPS, pp 648–654
57. Wan Q, An A (2006) An efficient approach to mining indirect associations. J Intell Inf Syst (JIIS) 27(2):135–158 (accepted)
58. Weka machine learning project. http://www.cs.waikato.ac.nz/ml/weka/index.html
59. Xiang Y (2002) Probabilistic reasoning in multiagent systems: a graphical models approach. Cambridge university, Cambridge
60. Xu W, Zhang X, Zhong J, Zhang W (2009) Attribute reduction in ordered information systems based on evidence theory. Knowl Inf Syst, published online 3
61. Zhou G, Su J (2003) Integrating various features in hidden markov model using constraint relaxation algorithm for recognition of named entities without gazetteers. In: Proceedings of IEEE international conference on NLP and KE, pp 732–739
62. Zhao J, Kan M, Procter P, Zubaidah S, Yip W, Li G (2004) Improving search for evidence-based practice using information extraction. In: Proceedings of the AMIA'10 annual symposium

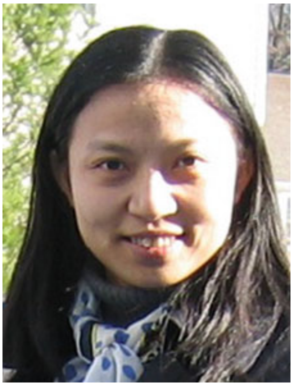
Author Biographies

Nick Cercone is Professor of Computer Science and Engineering and former Dean of the Faculty of Science and Engineering at York University. Cercone's research interests include natural language processing, knowledge-based systems, knowledge discovery in databases, data mining and design and human interfaces. He is the author of over 350 refereed publications and has graduated over 100 graduate students.

Xiangdong An is a postdoctoral fellow in the Department of Computer Science and Engineering at York University, Canada. He received his PhD in Computer Science from University of Waterloo, Canada. His research interests include uncertain reasoning, probabilistic graphical models, multi-agent graphical models, private data protection and information retrieval.



Jiye Li received her Bachelor degree in Computer Science from Northeastern University, China in 1999, and PhD degree in Computer Science from the University of Waterloo, Canada in 2007. She was a postdoctoral researcher at the Department of Computer Science and Engineering, York University in 2009. Her research interests include data mining, rough sets theory and medical data analysis.



Zhenmei Gu is a postdoctoral fellow in the Department of Computer Science and Engineering at York University, Toronto, Canada. She received her PhD degree in Computer Science from the University of Waterloo in 2006. She received her Master's degree in Computer Science from the University of Science and Technology of China in 1996 and her Bachelor's degree in Computer Science from Anhui University, China in 1993. Before she went into her PhD program in Canada, she worked as a software engineer at Apple Computer Inc., China for 2.5 years and at Sun Microsystems Ltd., Beijing, China for 1 year. Her research interests include information extraction and retrieval, data mining and natural language processing.

Aijun An is an Associate Professor in the Department of Computer Science and Engineering at York University, Toronto, Canada. She received her PhD Degree in Computer Science from the University of Regina. She held research positions at the University of Waterloo before joining York University in 2001. Her main research area is data mining. She has worked on various data mining problems, including classification, clustering, data stream mining, transitional and diverging pattern discovery, opinion mining, social network analysis and keyword search in graphs. She has published widely on premier journals and conferences in data mining, information retrieval and bioinformatics.