UNIVERSITY OF TWENTE.

Information Extraction, Data Integration, and Uncertain Data Management: The State of The Art

Technical Report

Authors: Mena B. Habib
Dr.ir. Maurice van Keulen
Faculty \ Department: EWI \ Database

Abstract

Information Extraction, data Integration, and uncertain data management are different areas of research that got vast focus in the last two decades. Many researches tackled those areas of research individually. However, information extraction systems should have integrated with data integration methods to make use of the extracted information. Handling uncertainty in extraction and integration process is an important issue to enhance the quality of the data in such integrated systems. This article presents the state of the art of the mentioned areas of research and shows the common grounds and how to integrate information extraction and data integration under uncertainty management cover.

Table of Contents

Information Extraction

Information Extraction (IE) systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. Domain-specific extraction patterns (or something similar) are used to identify relevant information. [1].

A typical IE system has basic phases for input tokenization, lexical analysis, name entity recognition, syntactical analysis, and identifying the interesting information required in a particular application [2]. Depending on the particular requirements of the application, IE systems may also include other modules. Figure 1 shows the modules that comprise a typical IE system.

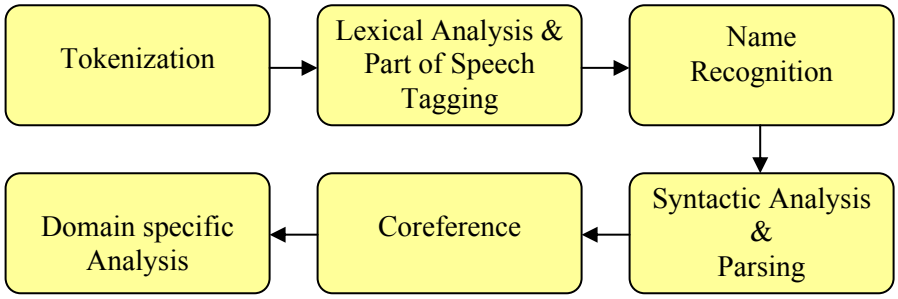

Figure 1: Modules for a Typical IE System

Tokenization phase identifies the sentences boundaries and splits each sentence into set of tokens. Splitting is performed along a predefined set of delimiters like spaces, commas, and dots. A token is a word or a digit, or a punctuation.

In the lexical analysis the tokens determined by the Tokenization module are looked up in the dictionary to determine their possible parts-of-speech and other lexical features that are required for subsequent processing. This module assigns to each word a grammatical category coming from a fixed set. The set of tags includes the conventional part of speech such as noun, verb, adjective, adverb, article, conjunct, and pronoun. Examples of well-known tag sets are the Brown tag set which has 179 total tags, and the Penn tree bank tag set that has 45 tags [3].

The next phase of processing identifies various types of proper names and other special forms, such as dates and currency amounts. Names appear frequently in many types of texts, and identifying and classifying them simplifies further processing. Furthermore, names are important for many extraction tasks. Names are identified by a set of regular expressions which are stated in terms of parts-of-speech, syntactic features, and orthographic features (e.g., capitalization). Personal names, for example, might be identified by a preceding title.

The goal of syntactic analyser is to give a syntactic description to the text. The analyser marks every word with a syntactic tag. The tags denote the subjects, objects, main verbs, etc. Identifying syntactic structure simplifies the subsequent phase of events extraction. After all, the arguments to be extracted often correspond to noun phrases in the text, and the relationships to be extracted often correspond to grammatical functional relations.

Given a text, relevant entities may be referred to in many different ways. Thus, success on the IE task is dependant on the success at determining when one noun phrase referred to the same entity as another noun phrase. A Co-reference module should handle the following problems:

- Name-alias co-reference. Names and their common variants must be recognized as coreferring, e.g., 'Silvia Miksch' and 'Prof. Miksch'.
- Pronoun co-reference. Pronouns like 'she', 'he', 'they', and so on must be associated with their reference.

The domain analysis is the core module of IE systems. The preceding modules prepare the text for the domain analysis by adding semantic and syntactic features to it. This module fills the templates. These templates consist of a collection of slots (i.e., attributes), each of which may be filled by one or more values.

There are a variety of approaches to construct this module. We can sum up those approaches mentioned in literatures into two main categories. The first is the inductive learning methods which are applied in IE to automatically learn rules that is useful for extraction tasks. Most of efforts focus on acquiring IE rules from a set of human annotated training documents. The second approach targeted using of common statistical machine learning methods for IE. Finally, we will present trials to handle uncertainty involved in IE process.

i. Rule-based IE

Rule-based IE systems consist of a set of linguistic rules. Those rules are represented as regular expressions or as zero or higher order logic.

Earlier researches developed these rules manually. However, due to the natural language richness with forms and contexts in which the information can appear, manual rules developing became very difficult and time consuming. Consequently, machine learning techniques are applied to automatically find the IE rules given a set of training corpora.

The basic idea behind the rule-based IE systems is that the system is fed with pairs of filled templates and annotated substrings associated to the filled slots in the template. Learning can be then viewed as a classification task where the extraction rules to be learned represent the conditions for filling a given slot or as pattern learning where the patterns are regular expression to be matched to text substrings.

Rule-based IE systems differ in many prospectives: i) the type of the text (structured, semi-structured, free text); ii) the type of learning (propositional,

relational); iii) the role of the user (classifying document as relevant or not, annotating sentences with a concept).

One of the earliest efforts was AutoSlog [4] which is a system that automatically builds a domain-specific dictionary of concepts for extracting information from text. As input, AutoSlog needs text in which the noun phrases that should be extracted have been labeled with domain-specific tags. For example, in a terrorism domain, noun phrases that refer to perpetrators, targets, and victims may be tagged. Given a tagged noun phrase and the original source text, AutoSlog first identifies the sentence in which the noun phrase appears. AutoSlog invokes a sentence analyzer called CIRCUS [5] to identify clause boundaries and syntactic constituents. AutoSlog determines which clause contains the targeted noun and looks up its list of heuristics rules (for samples see Table 1). Each rule generates an expression that likely defines the conceptual role of the noun phrase. In most cases, they assume that the verb determines the role. The rules recognize several verb forms, such as active, passive, and infinitive. An extraction pattern is created by instantiating the rule with the specific words that it matched in the sentence. As an example, consider the following sentence: "*Ricardo Castellar, the mayor was kidnapped yesterday by the FMLN*". Suppose that "Ricardo Castellar" was tagged as a relevant victim. AutoSlog passes the sentence to CIRCUS, which identifies *Ricardo Castellar* as the subject. Autoslog's subject heuristics are tested and the <subj> passive-verb rule fires. This pattern is instantiated with the specific words in the sentence to produce the extraction pattern <victim> was kidnapped. In future texts, this pattern will be activated whenever the verb "kidnapped" appears in a passive construction, and its subject will be extracted as a victim.

Table 1: Sample of AutoSlog heuristics.

Linguistic Pattern
<subject>passive-verb
<subject>active-verb
<subject>verb infinitive
<subject>auxiliary noun
passive-verb <direct-object>
active-verb <direct-object>
infinitive <direct-object>

An extension for AutoSlog system is AutoSlog-TS [6]. AutoSlog-TS generates extraction patterns using untagged text. The system needs only a pre-classified corpus of relevant and irrelevant texts. Nothing inside the texts needs to be tagged in any way. AutoSlog-TS operates exhaustively by generating an extraction pattern for every noun phrase in the training corpus. It then evaluates the extraction patterns by processing the corpus a second time and generating relevance statistics for each pattern. The process is illustrated in Figure 2.

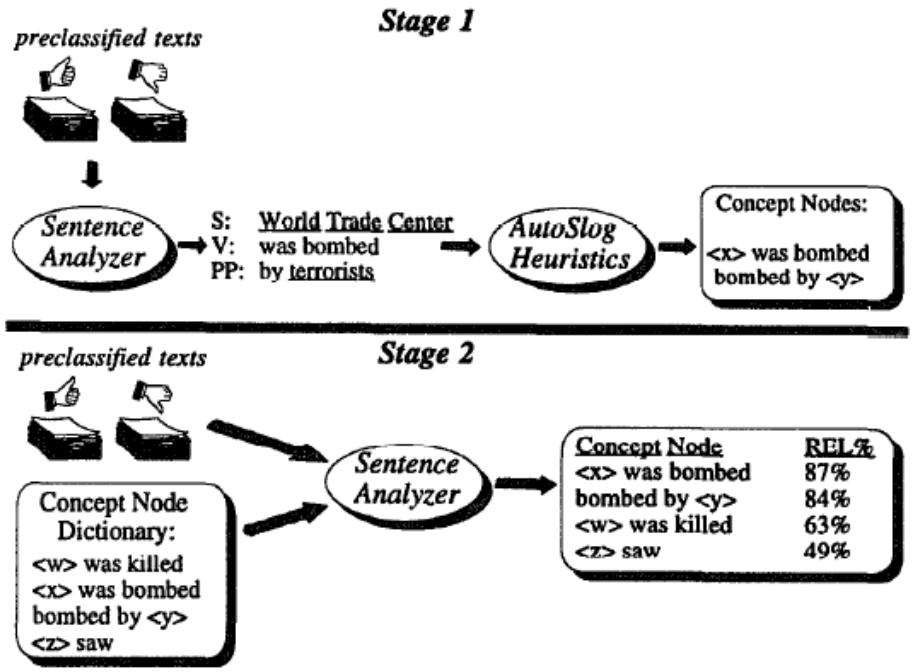

Figure 2: AutoSlog-TS flowchart

In Stage 1, the sentence analyzer identifies the noun phrases. For each noun phrase, the heuristic rules generate a pattern (concept node) to extract the noun phrase. AutoSlog-TS uses the set of heuristic rules used by AutoSlog plus two more. In Stage 2, the training corpus is processed a second time using the new extraction patterns. The sentence analyzer activates all patterns that are applicable in each sentence. A relevance statistics (relevance rate) for each pattern is then computed. More specifically, the conditional probability that a text is relevant given that it activates a particular extraction pattern is estimated by the formula:

$$\Pr(\text{relevant text} \mid \text{text contains patter } n_i) = \frac{\text{rel - freq}_i}{\text{total - freq}_i}$$

where rel - freq_i is the number of instances of pattern, that were activated in relevant texts, and total - freq_i is the total number of instances of pattern, that were activated in the training corpus. Next, the patterns are ranked in order of importance to the domain.

Another approach for IE is PALKA [7]. PALKA (Parallel Automatic Linguistic Knowledge Acquisition) is an IE system that acquires linguistic patterns from a set of domain-specific training texts and their desired outputs. It defines a specialized representation of patterns called FP-structures. Patterns are constructed in the form of FP-structures from training texts, and the acquired patterns are tuned further through the generalization of semantic constraints. Inductive learning mechanism is applied in the generalization step.

The linguistic pattern is represented as a pair of a meaning frame defining the types of information, and a phrasal pattern describing the syntactic ordering. This representation is called the FP-structure (Frame-Phrasal pattern structure) which is

similar to the AutoSlog's rules. The knowledge base is organized as a network of FP-structures and a concept hierarchy. Figure 3 shows an example of an FP-structure. A meaning frame is represented by a root, a set of slots, and semantic constraints on fillers. A phrasal pattern is an ordered combination of lexical entries or semantic categories. To combine a phrasal pattern and a meaning frame, each slot of the frame is linked to the corresponding element in the phrasal pattern.

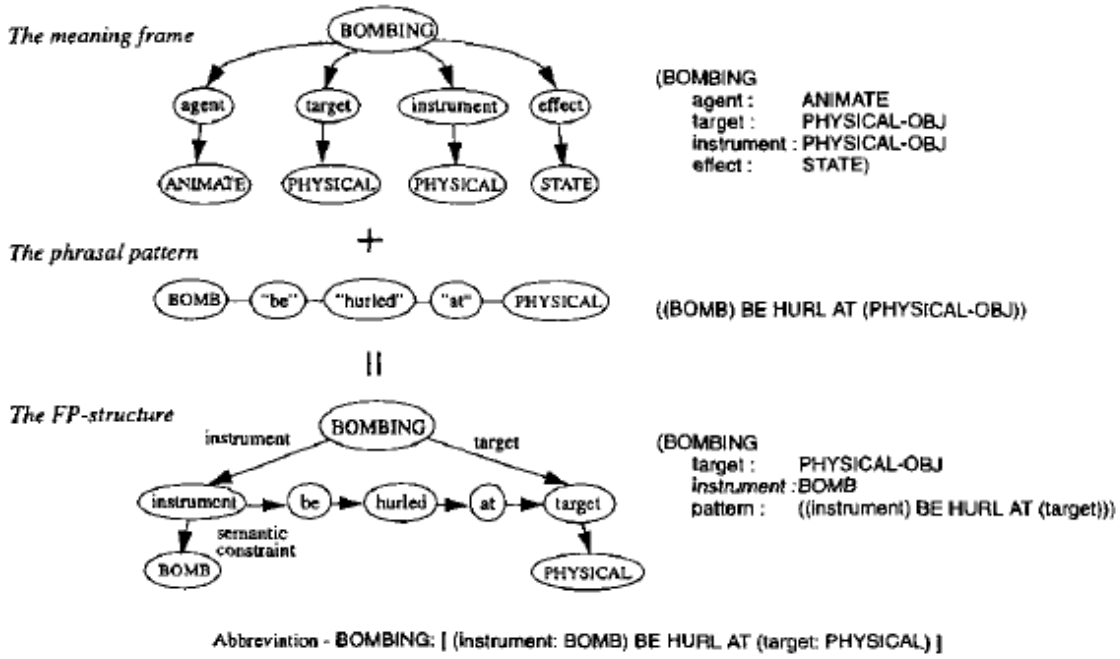

Figure 3: The frame-phrasal pattern representation

The FP-structures are used by the parser of the information extraction system to recognize input texts. By matching a phrase in the input text to the elements in a phrasal pattern, an FP-structure is activated, and by using the activated meaning frame, relevant information is extracted. Input words activate the phrasal pattern elements either directly or through the *isa* hierarchy of concepts.

Since the semantic category of a newly created pattern is determined to be the most specific one, it should be generalized if possible. An acquired FP-structure is compared with existing ones for generalization. Whenever two FP-structures with similar phrasal patterns are generated, their semantic constraints are generalized. When an over-generalized pattern is found (incorrect matching), the corresponding semantic constraint is specialized. Generalization and specification are done using semantic hierarchy until the resulting FP-structures cover all the initial specific ones.

Another approach that learns rules by generalizing specific ones is CRYSTAL [8] which is an IE system that automatically induces a dictionary of "concept-node (CN) definitions" sufficient to identify relevant information from a training corpus.

Similar to AutoSlog, CRYSTAL begins its induction with a dictionary of CN definitions built from each instance of the tagged training text. CRYSTAL uses a "specific-to-general" data driven search to find the most specific generalization that covers all positive and no negative instances. This is done by finding the number of relaxations required to unify two CN definitions. A new definition is then created with constraints relaxed just enough to unify the two definitions. Then CRYSTAL deletes from the dictionary all definitions covered by the new definition. One advantage for CRYSTAL is that it has the ability to extract multiple events mentioned in a single sentence.

A different approach that uses wordNet semantic hierarchy is the work done by Y. Chai and A. Biermann [9]. This system contains three major processes; address training; rule generalization; and the scanning of new information. WordNet is used in all three processes as shown in figure 4.

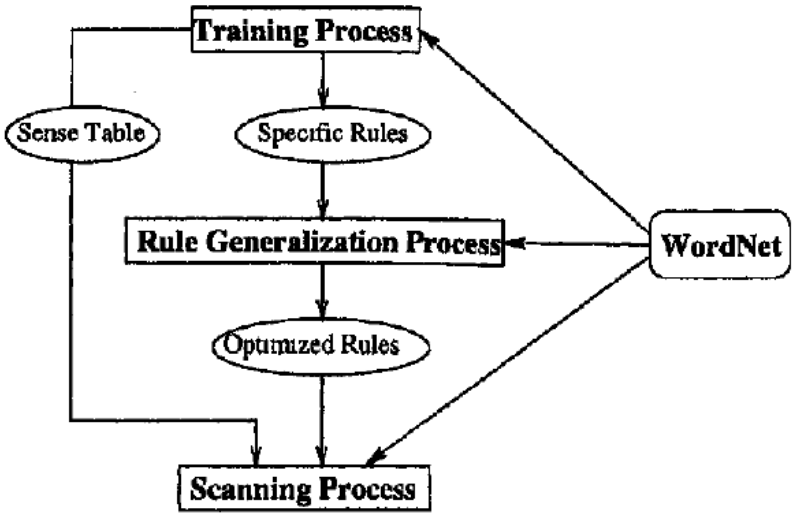

Figure 4: The use of WordNet in the system

WordNet is used to provide the sense information during the training process. The system assigns sense one to every headword of each noun/verb phrase, while providing the user the option to train the sense other than one. Rules created from the training process are specific to the training articles and must be generalized before being applied on other articles in the domain. Rules used in this system are represented as two parts left hand side (LHS) and right hand side (RHS). LHS is made up of the target objects in the form of noun phrases, and the verb or prepositional phrase indicating the relationship between the objects. The RHS of the rule consists of the operations required to create a semantic transition to add new object or to add new relation between objects.

The process of generalizing rules consists of replacing headwords in specific rules by a more general superordinate synset from its hypernym hierarchy in WordNet. This method keeps recall as high as possible by applying the most general rules, then the precision is adjusted by tuning the rules based on the user's specific inputs.

Another approach that represents rules in terms of first order logic is LIEP [10]. IE rules represented in LIEP like PROLOG predicates that represent relationship among extracted entities. Figure 5 shows an IE pattern used to identify a management change event.

```
n_was_named_t_by_c:
   noun-group(PNG,head(isa(person-name))),
   noun-group(TNG,head(isa(title))),
   noun-group(CNG,head(isa(company-name))),
   verb-group(VG,type(passive),head(named or elected or appointed)),
   preposition(PREP,head(of or at or by)),

   subject(PNG,VG),
   object(VG,TNG),
   post_nominal_prep(TNG,PREP),
   prep_object(PREP,CNG)
 ==> management_appointment(M,person(PNG),title(TNG),company(CNG)).
```

Figure 5: LIEP information extraction pattern

LIEP learns extraction patterns like that shown in figure 5 from example texts containing events. For each sentence of a training text given by the user, entities of interest (e.g. people, companies, and titles) are identified. LIEP tries to build a set of extraction patterns that will maximize the number of extractions of positive examples and minimize spurious extractions. Given a new example that is not already matched by a known pattern, LIEP first attempts to generalize a known pattern to cover the example. If generalization fails to produce a high-quality pattern, LIEP attempts to build a new pattern based on the example.

LIEP creates potential patterns from an example sentence/event by searching for sets of relationships that relate all of the role-filling constituents in the event to one another. These relationships between a pair of constituents can be either direct (like subject(A,B)) or indirect, where the constituents are the endpoints of a path of relationships that passes through other intermediate constituents (like subject(A,B), object(B,C)). LIEP uses recursive, depth-first algorithm to find relationships between pairs of constituents.

The new patterns LIEP learns are fairly specific: for non-role-filler constituents, they test for specific properties and head words. Often, later training examples have the same syntactic relationships as a previously learned pattern, but with different constituent head words or properties. This indicates that the pattern can be generalized. The system forms a generalization by inserting disjunctive values within each generalizable test in the pattern.

Inductive Logic Programming (ILP) is used in some IE researches. One of those researches is the robust automated production of information extraction rules

(RAPIER) [11]. RAPIER uses only a corpus of documents paired with filled templates to learn unbounded patterns that utilize limited syntactic information such as the output of a POS tagger and a lexicon with semantic classes, such as the hypernym links in WordNet without prior parsing or subsequent processing.

The extraction rules consist of three parts: 1) a pre-filler pattern that must match the text immediately preceding the filler, 2) a pattern that must match the actual slot filler, and 3) a postfiller pattern that must match the text immediately following the filler.

The Learning Algorithm consists of a specific to general (bottom-up) search. First, for each slot, most specific patterns are created for each example, specifying word and tag for the filler and its complete context. Given this maximally specific rule-base, RAPIER attempts to compress and generalize the rules for each slot. New rules are created by selecting two existing rules and creating a generalization.

WHISK [12] is an IE system that learns rules in the form of regular expressions that can extract either single slots or multiple slots.

```
Capitol Hill - 1 br twnhme. fplc D/W W/D. Undrgrnd pkg
incl $675. 3 BR, upper flr of turn of ctry HOME. incl gar,
grt N. Hill loc $995. (206) 999-9999 <br>
<i> <font size=-2> (This ad last ran on 08/03/97.)
</font> </i> <hr>
```

Figure 6a: An example of semi-structured text from an on-line rental ad.

```
ID:: 1
Pattern:: * ( Digit ) ' BR' * '$' ( Number )
Output:: Rental {Bedrooms $1} {Price $2}
```

Figure 6b: A WHISK rule to extract number of Bedrooms and Price.

```
Rental:
   Bedrooms:    1
   Price:       675

Rental:
   Bedrooms:    3
   Price:       995
```

Figure 6c: Output from the WHISK rule for Bedroom and Price.

Figures 6a, 6b, and 6c show a rental advertisement example, rule that is applied and the output slot respectively.

WHISK begins with a set of untagged instances and an empty training set of tagged instances. At each iteration of WHISK a set of untagged instances are selected and

presented to the user to annotate. The user adds a tag for each case frame to be extracted from the instance. The tags of training instances are used by WHISK to guide creation of rules and also to test the performance of proposed rules. If a rule is applied successfully to an instance, the instance is considered to be covered by the rule, else a new rule is created. WHISK induces rules top-down, first finding the most general rule that covers the seed, then extending the rule by adding terms one at a time. Top-down rule induction typically begins with an "empty" rule (like " * (*) * (*) * (*) * ") that covers all instances, then adds terms to the rule, which reduces the number of instances covered monotonically. This is done by adding tokens within the slot-filler boundaries. The growth of a rule continues until it covers the training set.

A step towards unsupervised learning is Dual Iterative Pattern Relation Expansion (DIPRE) [13]. DIPRE uses the bootstrapping learning approach which is a technique used to iteratively improve the IE system performance. It addresses the problem of extracting a relation of books (author, title) pairs from the Web. It begins with a small seed set of (author, title) pairs then it finds all occurrences of those books on the Web. From these occurrences patterns for the citations of books are recognized. Then it searches the Web for these patterns and finds new books and so on. A pattern in DIPRE is composed of five string fields: *prefix category1 middle category2 suffix*. A text fragment matches the pattern if it can be split to match each field. For instance, to learn the relation (Author, Book title) from Web pages, DIPRE learned the pattern '<LI><B>*title*</B> by *author* (', where the text preceding the title is the prefix, the text between the title and the author is the middle, and the suffix consists of the text following the author.

Another similar research is ExDISCO [14]. ExDISCO identifies a set of relevant documents and a set of event patterns from un-annotated text, starting from a small set of seed patterns. Starting with a large corpus of documents in the domain and an initial seed of scenario patterns selected by the user, the pattern set is used to divide the corpus into a set of relevant and irrelevant documents. These patterns are in the form of Subject-Verb-Object. Each document in the corpus is automatically converted into a set of candidate patterns, one for each clause. Patterns are then ranked by the degree to which their distribution is correlated with document relevance. The highest ranking patterns are added to the pattern set. The new pattern set is used to induce a new split of the corpus into relevant and irrelevant documents. The procedure is repeated until some iteration limit is reached, or no more patterns can be added.

ii. Statistical-based IE

Although rule-based IE techniques are the most common ones used in IE, many approaches explore the use of statistical machine learning methods with this area. These methods include Hidden Markov Models (HMM), Maximum Entropy Models (MEM), Conditional Random Fields (CRF), and Support Vector Machines (SVM).

Hidden Markov Models

HMM is a powerful tool that has been applied to wide range of applications in language processing field like part-of-speech tagging and speech recognition. One of the earliest researches for using HMM applied to IE area was that done by [15]. In this work, the HMM state transition probabilities and word emission probabilities are learned from labeled training data. The HMM has background and target states. Target states model the text of interest. Other states model background language, including the prefix and suffix of the target state. HMM is learned from annotated documents. Annotations label each word as a target or a non-target word. The Viterbi algorithm is used to extract information from documents modelled by a HMM. This paper overcame the problem of sparse training data which causes poor probability estimates (i.e. unseen words have emission probabilities of zero) by using Shrinkage smoothing. Shrinkage combines the benefit of having robust probability estimates of simple HMM and the benefit of learning concepts more precisely of complex HMM. Shrinkage is typically defined in terms of some hierarchy that represents the expected similarity between parameter estimates, with the estimates at the leaves. Figure 7 shows such a hierarchy. It depicts, for example, that all prefix states are expected to have related word distributions reflecting also the fact that in a simpler model, all four prefix states might have been represented by a single state that allowed up to four self-transitions.

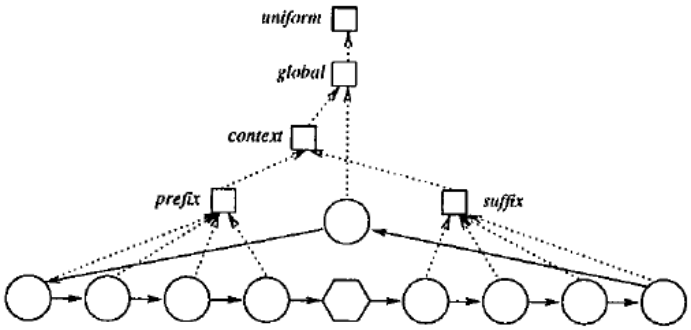

Figure 7: A shrinkage configuration that addresses data sparsity in contextual states, showing shrinkage only for non-target states (circular states).

Shrinkage-based parameter estimate in a leaf of the hierarchy (state of the HMM) is a linear interpolation of the estimates in all distributions from the leaf to its root. Local estimates are calculated from their training data by maximum likelihood. The training data for an internal node of the hierarchy is the union of all the data in its children.

In an extension to their previous work [16], the authors demonstrates that extraction accuracy strongly depends on the selection of structure, and presents an algorithm for automatically finding good structures by stochastic optimization. Their method begins with a minimal number of states, explores various state splitting operations, selects the operation that gives best performance on a labeled validation set, and

recursively explores further splitting operations. The final model is then chosen by cross-validation from those generated. Beginning with the simple model, a hill-climbing in the space of possible structures is performed. At each step, each of a set of operations is applied to the current model and selecting one of the resulting structures as the next model. For the experiments reported they defined seven operations (Lengthen a prefix, Split a prefix, Lengthen a suffix, Split a suffix, Lengthen a target string, Split a target string, and Add a background state). The selected model in each step is the one from the series of generated structures that scores the best F1 measure in separate runs of three-fold cross-validation on the training set.

On the same context Seymore et al [17] explored the use of hidden Markov models for information extraction tasks, specifically focusing on how to learn model structure from data and how to make the best use of labeled and unlabeled data. They introduced the concept of distantly-labeled data, which is labeled data from another domain whose labels partially over-lap those from the target domain. They proved that a manually-constructed model that contains multiple states per extraction field outperforms a model with one state per field. Training data labeled with class information can be used to build a maximally-specific model. This model can be used as the starting point of a variety of state merging techniques. Two simple types of merges that can be used to generalize the maximally-specific model are proposed. First, "neighbour-merging" combines all states that have the same class label. Second, "V-merging" merges any two states that have the same label and share transitions from or to a common state. Once the model structure has been selected, the distantly-labeled data is used along with labeled and unlabeled data to estimate the transition and emission parameters.

Another approach considering HMM in IE task presented by [18]. In this work IE is applied on free text in biomedical domain to extract *n*-ary relations (multiple slot). The authors investigate an approach to incorporating information about the grammatical structure of sentences into HMM architectures. Also they investigate an objective function for HMM training whose emphasis is on maximizing the ability of the learned models to identify the phrases of interest rather than simply maximizing the likelihood of the training data. They adopt a training method that is designed to maximize the probability of assigning the correct labels to various parts of the sentences being processed. Their approach is based on using syntactic parses of all sentences they process. Each phrase segment consists of a type describing the grammatical nature of the phrase, and the words that are part of the phrase. The sentence is flattened into a sequence of phrase segments. The states in the HMMs represent the annotated segments of a sentence. A given state can emit only segments whose type is identical to the state's type. Modified versions of the standard Forward, Backward and Viterbi algorithms are used in training and testing.

Maximum Entropy Models

The maximum entropy (ME) framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints

imposed. ME model can easily combine diverse features. These features can be quite complex and allow the experimenter to make use of prior knowledge about what types of information are expected to be important for classification. It is widely adopted in many natural language processing and text mining tasks, such as Part of Speech (POS) tagging, named entity recognition and relation extraction.

Maximum entropy is used by [19] in extraction of a single-slot information from semi-structured text (Seminar Announcements) as well as of a multi-slot from free text domain (Management Succession). The authors used ME to classify words into slots. The preceding and succeeding words to the tagged word in the training set are used for classification. To overcome the problem of sequence of inadmissible classes, a transition probability is defined between word classes and the Viterbi algorithm is then used to select the sequence of word classes with the highest probability. In the multi-slot domain, the ME is used to classify the relations between slots element either to be positive (existing) relation or negative (non existing) relation.

In a similar approach [20], ME is used for predicting the type of relation between every pair of entities within each sentence. For each pair of mentions, several feature streams (words, entity type, overlap, dependency, parse tree, etc.) are computed. All the syntactic features are derived from the syntactic parse tree and the dependency tree that is computed using a statistical parser. ME is trained using the mentioned set of features.

A different approach [21] used maximum entropy Markov models (MEMMs) to address the problems of traditional HMM. According to this paper traditional approaches of HMM have two problems. The first is that it ignores richer representation of observations that describes observations in terms of many overlapping features, such as capitalization, word endings, part-of-speech, formatting, position on the page, and node memberships in WordNet. The second problem is that it considers independent observations on each other and only dependant on current starte. In the proposed maximum entropy Markov models (MEMMs), the HMM transition and observation functions are replaced by a single function that provides the probability of the current state given the previous state and the current observation. Generalized Iterative Scaling (GIS) is used for parameter estimation.

Conditional Random Fields

A Conditional Random Field (CRF) [22] is an undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. Conditional random fields offer set of advantages: discriminative training models; different types of features; efficient training based on dynamic programming; and parameter estimation guaranteed to find the global optimum. An application of CRF to the problem of IE is the system proposed by [23].

Support Vector Machines

Support Vector Machines (SVM) are widely used in classification problems. However, several attempts to apply them to IE tasks have been made. To do this, it is required to represent the IE problem as a classification problem. Once it has been transformed into a classification problem, other ML methods can be applied.

First well known trial of using SVM with IE is that done by [24]. On this work the authors argued that traditional pattern templates methods restrict the kind of extraction patterns that can be learned by IE systems. To allow a wider range of context information to be considered in learning extraction patterns, they proposed to model the content and context information of the candidate entity to be extracted as a set of features. A classification model is then built for each category of entities using Support Vector Machines (SVM). Their learning process consists of three phases. The first is to parse all the sentences in the document. Only the simple noun-phrases are selected as candidate entities which are grouped as positive entities in case their corresponding noun-phrases match the information to be extracted. The rest are used as negative entities. The second phase is to drive features (content and context) for the training target entities. The final step is to construct the extraction model using SVM.

Another similar approach is the ELIE system [25]. the approach that is used in this system is to treat the identification of fragment start and end positions as distinct token classification tasks. The instances are all tokens in the document. All tokens that begin a labeled field are positive instances for the start classifier, while all the other tokens become negative instances for this classifier and similarly with the end classifier. The features include the specific token, part-of-speech (POS), chunking, orthographic and gazetteer information. The system also investigates how different feature-sets contribute to the performance. The system is enhanced by using convergent boundary classification which combines the results of two sets of classifiers, one set with high precision (Level One (L1) learning) and one with high recall (Level two (L2) learning). The L1 learner treats IE as a standard classification task, augmented with a simple mechanism to attach predicted start and end tags. As the L1 learner builds its model based on a very large number of negative instances and a small number of positive instances, it is more likely to produce false negatives than false positives. The L2 learner is trained to detect either the end of a fragment given its beginning, or the beginning of a fragment given its end. The L2 models are likely to have much higher recall but lower precision. Figure 8 shows the extraction task using the two levels models.

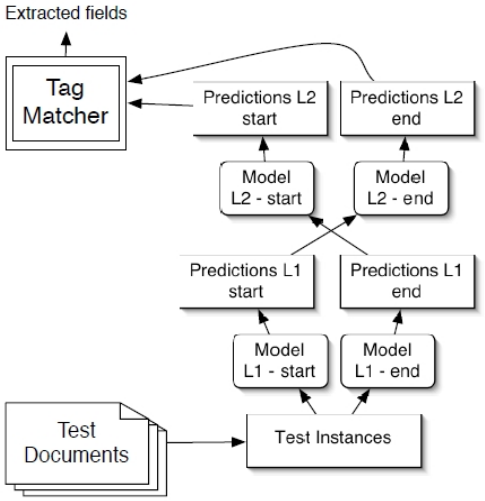

Figure 8: The extraction task in the ELIE

Another approach that uses SVM in relation extraction is that done by [26]. This work presents a relation detection approach that combines clues from three levels of syntactic processing (tokenization, sentence parsing and deep dependency analysis) using kernel methods. Each source of information is represented by kernel functions. Then composite kernels are developed to integrate and extend individual kernels so that processing errors occurring at one level can be overcome by information from other levels.

Data Integration

i. Integration of Extracted Information

The common definition for data integration (DI) is the systems that provide a unified view of data stored in a number of different sources. The data sources, each with an associated schema, are integrated to form a single database with a global schema.

However, in information extraction systems, data integration has different definition which is the process of integrating the extracted information with those already existing in the databases. This process is divided into two steps. The first step is the duplication detection which aims to decide if two objects refer to the same entity in spite of the many noisy variants which appear in the unstructured source. The second step is data fusion in which the duplicate representations are combined into a single representation while inconsistencies in the data are resolved. A quite a few approaches and techniques address the problem of conflicts resolving when integrating data. Data integration for information extraction systems should make use of the duplicate entry to modify the certainty factor of the extracted piece of information. Within this context, existing approaches of data integration for information extraction can be divided into three ways depending on the sequence in which the extraction and integration tasks are ordered [27].

1) Decoupled Extractions and Integration: In this scenario, extraction task is performed independently of data integration steps and independent of the extraction of other pieces of information. The task of integration is to decide if a given extracted template is the same as any of the existing database entries, or if it is a new record. This problem called de-duplication problem which can be seen as the task of assigning a boolean value to each pair (*r*, *e*), where *r* is the extracted record and *e* is each entry in the existing database. The Boolean value indicates whether or not *r* is a duplicate of *e*. If the Boolean value was *true* for one or more database entries then the extracted record *r* should be integrated with the best matching entry *e*, otherwise *r* must be inserted as a new entry.

This assignment process can be performed as a simple classification task where the input features to the classifier are string similarity functions such as cosine similarity, or edit distance. The classifier itself can be either a set of manually defined rules on the similarity functions or an automatically trained classifier like SVM or a decision tree.

2) Decoupled Extraction and Collective Integration: In this approach, all extracted frames are collectively grouped first before being integrated, instead of integrating one entry at a time. As an example, consider three frames with single field extracted in the order F1, F2, F3.

F1. Alistair MacLean
F2. A Mclean
F3. Alistair Mclean

After the extraction of F1, F2, it is unlikely that F1 will be classified as a duplicate of F2 because the last names of F1 and F2 do not match exactly and the first name is abbreviated. However, after extracting F3, F2 and F1 start looking similar because of the transitivity of the duplicate-of relation.

The same approach can be used in multiple-fields frames by performing simultaneous inference for all candidate pairs, and allowing information to propagate from one candidate match to another via the attributes they have in common [28].

3) Coupled Extraction and Integration: In this approach, the two steps are performed jointly instead of independently to improve accuracy. It is believed that there is little to be gained out of joint extraction and integration when the database is not guaranteed to be complete. An example for this approach is [29], which showed how to integrate unstructured text records into existing multi-relational databases using models that combine clues from both existing entities in the database and labelled unstructured text. It extends semi-Markov CRFs to capture pattern-level and entity-level information in the database, and uses these to extract entities and integrate them in a database.

ii. Duplication Detection

Duplication detection mainly is the problem of similarity matching [30]. If pair of objects has a similarity measure greater than a predefined threshold, then this pair is considered a duplicate. All duplication detection approaches focus on the problems of efficiency and effectiveness.

Effectiveness is related to the quality of the similarity measure and the choice of the threshold. Some similarity measures are domain dependant and some others are not, like the Levenshtein string distance measure. The choice of the threshold also is problematic as high threshold value will leed to high precision but poor recall and vice versa with the low threshold value.

Efficiency also is an issue in case of huge datasets. Calculating similarity among all pairs of objects is a complex task. More complexity comes from the string similarity measure itself.

The result of the duplicate detection process is to generate the duplicate clusters which contain all representations of the same object. The goal of data fusion is to fuse these multiple representations into a single one.

iii. Conflict Classification

As discussed before, data integration system should resolve the conflict among the data of the extracted frames. A data conflict is present if, for the same real-world object (e.g., a hotel room price), semantically equivalent attributes, from one or more sources, do not agree on its attribute value (e.g., extracted frame 1 reporting "$123" as the average room price, while extracted frame 2 reporting "$154").

According to [30], there is two kinds of data conflict: (a) uncertainty about the attribute value, caused by missing information; and (b) contradictions, caused by different attribute values.

1) Uncertainties: An uncertainty is a conflict between a non-null value and one or more null values that are all used to describe the same object. This is caused by missing information, for example, null values in the extracted frame, or an attribute completely missing in an extracted frame.

2) Contradictions: A contradiction is a conflict between two or more different non-null values that are all used to describe the same object. This is the case if two or more extracted frames provide two or more different values for the same attribute on the same object.

iv. Data Fusion

There are several simple strategies to handle inconsistencies. According to [31], those strategies can be classified as shown in figure 9 into three main classes based on the way they handle conflicting data: ignorance, avoidance, and resolution.

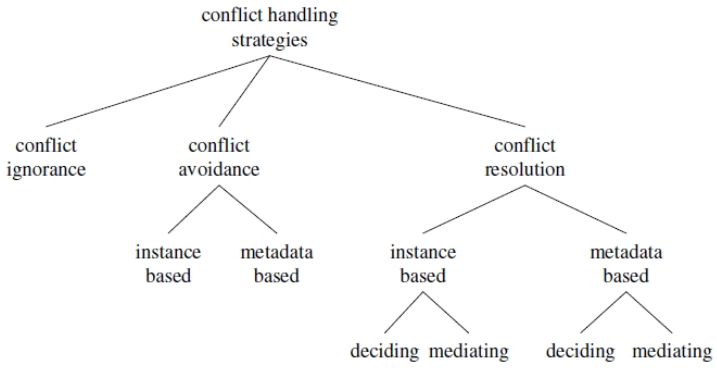

Figure 9: A classification of strategies to handle inconsistent data

1) Conflict Ignorance: describes strategies that do not make a decision with respect to conflicts at all or sometimes not even aware of the conflict occurrence. Two approaches are the *Pass It On* and the *Consider All Possibilities* strategy. *Pass It On* strategy passes all conflicting values to the user or another application and lets the user or application decide how to handle possible conflicts among the values. While *Consider All Possibilities* strategy tries to be as complete as possible by

enumerating all possibilities and giving the user the choice among all possible combinations of values.

2) Conflict Avoidance: Those strategies are aware of conflicts but do not perform individual resolution for each conflict. Rather, a single decision is made, like preference of a source, and applied to all conflicts. This class can further be divided into two classes, one that takes metadata into account when taking a decision (metadata based) and one that does not (instance based). Two examples for the instance based approach are *Take The Information* and *No Gossiping*. *Take The Information* approach prefers to consider non-null values over the null values while *No Gossiping* approach leaves inconsistencies out and report only on the certain facts. An example for the metadata based approach is *Trust Your Friends*. The idea behind this strategy is to trust a third party to either provide the correct value or the correct strategy. Whom to trust is decided once and carried out for all data values. The source preference can be decided by the user, but this can also be done automatically by choosing the most reliable or the largest source or by using other quality criteria.

3) Conflict Resolution: This strategy tries to give a fusion decision for each individual conflict. Such decisions can be *instance-based* which consider the actual conflicting data values, or *metadata-based* which choose values based on metadata, such as freshness of data or the reliability of a source. Those strategies can be subdivided according to the result they are able to produce: *deciding* strategies choose a preferred value among the existing values, while *mediating* strategies can produce an entirely new value, such as the average of a set of conflicting numbers. Examples for i*nstance-based*, *deciding* approach are the *Cry With The Wolves*, which selects the most frequent value, and *Roll The Dice* which just selects a random value among the inconsistent values. An example of an *Instance-based*, *mediating* strategy is *Meet In The Middle* which computes and uses the average of the inconsistent values. Finally an example for *Metadata-based*, *deciding* approach is *Keep Up To Date* which uses the most recent value and requires some additional time-stamp information.

Uncertainty in Information Extraction and Data Integration

Imprecision in information extraction is expected, especially in unstructured text where a lot of noise exists. There is an increasing research interest in more formally handling the uncertainty of the extraction process so that the answers of queries can be improved with quality measures. Only recently has information extraction and probabilistic database research intersected [32][33]. We will focus on the problem of how to populate a probabilistic database to represent the uncertainty of extraction.

Let us consider the simplest scenario of extractions of some structured entity. Imprecision in information extraction can be represented by associating each extracted information with a probability value. Other methods method extends this approach to output multiple possible extractions instead of a single extraction. It is easy to extend probabilistic models like HMM and CRF to return the *k* highest probability extractions instead of a single most likely one and store them all into the probabilistic database [34]. Figure 10 shows an example for the output of an extraction of book titles from sentences [27].

Id	Title	Pr
1	Last Theorem	0.5
1	Fermat's Last Theorem	0.3
1	The Fermat's Last Theorem	0.2
2	"Transaction Processing: Concepts and Techniques"	0.95
2	Transaction Processing	0.05

Figure 10: Example for the output of an extraction of book titles from sentences

Managing uncertainty in rule-based approaches is more difficult than in the statistical ones. In rule-based systems each rule is associated with a precision value that indicates the percentage of cases where the action associated with that rule is correct. However, there is little work on maintaining probabilities when the extraction is based on many rules, or when the firings of multiple rules overlap. Within this context, [34] presents a probabilistic framework for managing the uncertainty in rule-based information extraction systems where the uncertainty arises due to the varying precision associated with each rule by producing accurate estimates of probabilities for the extracted annotations. They also capture the interaction between the different rules, as well as the compositional nature of the rules.

The case of extracting multiple attributes differs from the case of extracting single ones as we cannot assume that the different attributes extracted are independent of each other. The results of these multiple attributes extractions are stored as multiple columns table. A simple extension of the previous multi-row imprecision model is to maintain with each row a probability value exactly as in the single column case. Another approach of representing uncertainty is through a probability distribution attached to each column. In this approach each column stores a probability distribution of the possible values that it can take. The probability of any unique value is computed by multiplying the probability

of the corresponding values from each column. These probabilities are more abstract than storing the probability for each possible value at the row-level. A third approach presented in [33] investigate a model that captures both row-level and column-level uncertainty and show that this representation provides significantly better approximation compared to models that use only row or only column level uncertainty. Figures 11a, 11b, and 11c shows an example for the three approaches (row, column, and hybrid) of representing uncertainty for multi-attribute extraction [27].

Id	House_no	Area	City	Pincode	Prob
1	52	Goregaon West	Mumbai	400 062	0.1
1	52-A	Goregaon	West Mumbai	400 062	0.2
1	52-A	Goregaon West	Mumbai	400 062	0.5
1	52	Goregaon	West Mumbai	400 062	0.2

Figure 11a: Row representation of uncertainty for multi-attribute extraction

Id	House_no	Area	City	Pincode
1	52 (0.3)	Goregaon	Mumbai (0.6)	400 062
	52-A (0.7)	West (0.6)	West Mumbai	(1.0)
		Goregaon (0.4)	(0.4)	

Figure 11b: Column representation of uncertainty for multi-attribute extraction

Id	House_no	Area	City	Pincode	Prob
1	52 (0.167)	Goregaon	Mumbai (1.0)	400 062	0.6
	52-A (0.833)	West (1.0)		(1.0)	
1	52 (0.5)	Goregaon (1.0)	West	400 062	0.4
	52-A (0.5)		Mumbai (1.0)	(1.0)	

Figure 11c: Hybrid representation of uncertainty for multi-attribute extraction

As the information extracted from multiple text sources may contain redundant information, probabilistic data integration efforts are needed to integrate and manage uncertainty within the extracted frames. In this case, two kinds of uncertainties need to be handled. The first is the single value uncertainty represented by the probabilities in each row. The second is the uncertainty comes from duplicate detection measures and if a pair of extracted objects refers to the same entity or not. Combining those kinds of uncertainties is a challenging problem. Current approaches either ignore the first or the second kind of uncertainty and deals only with the other type.

Within the first set of approaches that considers only the duplicate detection and the co-reference uncertainty, [36] considers the problem of citation matching—the problem of deciding which citations correspond to the same publication. This approach explains how to specify a generative probability model of this domain. It handles identity uncertainty by incorporating probabilities over the possible mappings from terms in the language to objects in the domain. Another work done in this direction is [37]. This paper introduces several discriminative, conditional probability models for co-reference analysis. It uses the relational models that incorporate a great variety of features of the input without having to be concerned about their dependencies.

On the other hand, the other set of approaches assumes only extraction uncertainty and ignore co-reference uncertainty. It aims to combine multiple extractions uncertainty values into a one [27].

One example for this approaches called *The Noisy-OR Model*. To describe this model consider the n repetitions for an extracted string x with different probabilities $p_1, \ldots, p_n$. The objective of this approach is to combine those probabilities into a single probability value P of x being a correct object of the entity y. The simplest approach is to assume that all extractions are independent of each other. The probability that x is of type y is calculates as follow:

$$P = 1 - \prod_{i=1}^{n} (1 - p_i)$$

This formula has a draw back of assigning high probability to extracted string x that is repeated a high number of times with low probability (ex: 100 repetitions with probability 0.1 will give a combined probability near 1). A modified version of this approach called *soft-OR* tries to overcome this problem by selecting the maximum of the existing precisions as follow:

$$P = \max_{i=1}^{n} p_i$$

The problem with the *soft-OR* function is that it does not reward repetitions. Two other directions assume that there are multiple rules that can extract an object for an entity. Let $M_1,\ldots,M_k$ denote these rules. Let $n_1,\ldots,n_K$ denotes the number of times string x was extracted using rules $M_1,\ldots,M_k$ respectively. The objective of those directions is to estimate the probability that x is indeed an instance of type y, $P(x \in y | n_1, \ldots, n_K)$ either surprisingly or unsurprisingly.

The supervising approach called *conditional probability models from labelled data*. It uses well-known methods such as logistic classifier as follows:

$$P(x \in y | n_1, \ldots, n_K) = \frac{1}{1 + \exp(\sum_{i=1}^{k} w_i n_i + b)}$$

Where the parameters $w_1,\ldots,w_K$ and b can be estimated using training data during the training phase.

The unsupervised approach *generative models for unlabeled data [38]*. This approach introduces a combinatorial model called "balls and urns". This model computes the impact of sample size, redundancy, and corroboration from multiple distinct extraction rules on the probability that an extraction is correct. It describes methods for estimating the model's parameters in practice.

Many other directions of research are done to handle imprecision in extraction systems. One of those directions is concerned with querying the results of uncertain extractions [34][39]. Another direction that tries to model imprecision and uncertainty in information extraction using fuzzy set models and functions discussed in [40]. One more direction is

modelling uncertainty of extracted events using context [41]. It uses linguistic patterns for uncertainty and reality detection.

In conclusion, existing approaches managing imprecision in information extraction is still at a preliminary stage. More efforts are needed to manage all kinds of imprecision together to create reliable models that are more practically useful.

References

[1] Marie-Francine Moens. “Information Extraction: Algorithms and Prospects in a Retrieval Context”. The Information Retrieval Series, Springer. 2006.

[2] K. Kaiser, S. Miksch. "Information Extraction: A Survey". Scientific Report, Vienna University of Technology, Institute of Software Technology & Interactive Systems, Asgaard-TR-2005-6. 2005.

[3] C. D. Manning and H. Schütze. “Foundations of Statistical Natural Language Processing”. Cambridge, The MIT Press. 1999.

[4] E. Riloff. “Automatically Constructing a Dictionary for Information Extraction Tasks”. Proc. Of the 11th national conference on artificial intelligence. Pages 811-816. 1993.

[5] W. Lehnert. “Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds”. Advances in Connectionist and Neural Computation Theory, Vol. 1. Pages 135-164. 1990.

[6] E. Riloff. “Automatically generating extraction patterns from untagged text”. In Proceedings of the Thirteenth National Conference on Artificial Intelligence. Pages 1044–1049. 1996.

[7] J. Kim and D. I. Moldovan. “Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction”. IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 5. Pages 713-724. 1995.

[8] S. Soderland, D. Fisher, J. Aseltine, W. Lehnert. “CRYSTAL Inducing a Conceptual Dictionary”. In Proceedings of the 14th International Joint Conference on Artificial Intelligence. Pages 1314-1319. 1995.

[9] J. Y. Chai and A. W. Biermann. “The Use of Lexical Semantics in Information Extraction”. In Proceedings of the ACL Workshop on Natural Language Learning. 1997.

[10] S. B. Huffman. “Learning Information Extraction Patterns from Examples”. Lecture Notes in Computer Science, Vol. 1040. Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing. Pages 246-260. 1996.

[11] M. E. Califf and R. J. Mooney. “Relational Learning of Pattern-Match Rules for Information Extraction”. In T.M. Ellison (ed.) CoNLL97: Computational Natural Language Learning. Pages 9-15. 1997.

[12] S. Soderland. “Learning Information Extraction Rules for Semi-Structured and Free Text”. Machine Learning, Vol. 34, Issue 1-3. Pages 233-272. 1999.

[13] S. Brin. “Extracting patterns and relations from the world wide web”. In WebDB Workshop at 6th International Conference on Extending Database Technology, (EDBT’98). Pages 172-183. 1998.

[14] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. “Automatic acquisition of domain knowledge for Information Extraction”. In the proceedings of the 18th conference on Computational linguistics - Vol 2. Pages 940 – 946. 2000.

[15] D. Freitag , and A. K. Mccallum. “Information Extraction with HMMs and Shrinkage”. In Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction. Pages 31-36. 1999.

[16] D. Freitag , and A. Mccallum. “Information Extraction with HMM Structures Learned by Stochastic Optimization”. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. Pages 584-589. 2000.

[17] K. Seymore , A. Mccallum , and R. Rosenfeld. “Learning Hidden Markov Model Structure for Information Extraction”. In Proceedings of the AAAI 99 Workshop on Machine Learning for Information Extraction. Pages 37-42. 1999.

[18] S. Ray , and M. Craven. “Representing Sentence Structure in Hidden Markov Models for Information Extraction”. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence. Pages 1273-1279. 2001.

[19] H. L. Chieu , and H. T. Ng. “A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text”. In Proceedings of the Eighteenth National Conference on Artificial Intelligence. Pages 786-791. 2002.

[20] N. Kambhatla. “Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations”. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL). 2004.

[21] A. McCallum, D. Freitag, and F. C. N. Pereira. “Maximum entropy Markov Models for Information Extraction and Segmentation”. In Proceedings of the Seventeenth International Conference on Machine Learning. Pages 591-598. 2000.

[22] J. Lafferty, A. McCallum, F. Pereira. “Conditional random fields: probabilistic models for segmenting and labeling sequence data”. In Proceedings of the 18th International Conference on Machine Learning (ICML). Pages: 282-289. 2001.

[23] C. Cox, J. Nicolson, J. Finkel, C. Manning, P. Langley. “Template Sampling for Leveraging Domain Knowledge in Information Extraction”. First PASCAL Challenges Workshop. 2005.

[24] A. Sun , M. Naing , E. Lim, and W. Lam. “Using Support Vector Machines for Terrorism Information Extraction ”. In Proceedings of the First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI03). Pages 1–12. 2003.

[25] A. Finn and N. Kushmerick. “Information Extraction by Convergent Boundary Classification”. In Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining. 2004.

[26] S. Zhao, R. Grishman. “Extracting Relations with Integrated Information Using Kernel Methods”. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Pages: 419-426. 2005.

[27] S. Sarawagi. “Information Extraction”. In Foundations and Trends in Databases, Vol. 1, No. 3. Pages 261–377. 2008.

[28] P. Singla and P. Domingos. “Multi-relational record linkage”. In Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining. Pages: 31-48. 2004.

[29] I. Mansuri and S. Sarawagi. “A system for integrating unstructured data into relational databases”. In Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE). Pages 29-40. 2006.
[30] J. Bleiholder, and F. Naumann. “Data Fusion”. ACM Computing Surveys, Vol 41, No. 1. Pages 1:1 - 1:41. 2008.
[31] J. Bleiholder, and F. Naumann. “Conflict Handling Strategies in an Integrated Information System”. In Proceedings of the International Workshop on Information Integration on the Web (IIWeb). 2006.
[32] M. Wick. “Representing Uncertainty in Databases with Scalable Factor Graphs”. Masters Thesis/Synthesis. 2009.
[33] R. Gupta, and S. Sarawagi. “Creating Probabilistic Databases from Information Extraction Models”. In Proceedings of the 32nd international conference on very large data bases (VLDB 2006). Pages 965-976. 2006.
[34] D. Z. Wang, M. J. Franklin, M. Garofalakis, and Joseph M. Hellerstein. “Querying probabilistic information extraction”. In Proceedings of the 36th international conference on very large data bases (VLDB 2010). 2010.
[35] E. Michelakis, R. Krishnamurthy, P. Haas, and S. Vaithyanathan. “Uncertainty Management in RuleBased Information Extraction Systems”. In Proceedings of the 35th SIGMOD international conference on Management of data. Pages 101-114. 2009.
[36] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. “Identity uncertainty and citation matching”. In Advances in Neural Information Processing Systems (NIPS). 2002.
[37] A. McCallum and B. Wellner. “Toward conditional models of identity uncertainty with application to proper noun Co-reference”. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web. Pages 79-86. 2003.
[38] D. Downey, O. Etzioni, and S. Soderland, “A probabilistic model of redundancy in information extraction”. In Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-05. Pages 1034-1041. 2005.
[39] N. N. Dalvi and D. Suciu, “Efficient query evaluation on probabilistic databases”. In Proceedings of the 36th international conference on very large data bases (VLDB 2004). Pages 864-875. 2004.
[40] I. B. Turksen and A. Celikyilmaz. “Information Extraction From Text – Dealing With Imprecise Data”. Fuzzy Systems, Book Chapter, In-TECH. 2010.
[41] B. Goujon. “Uncertainty Detection for Information Extraction”. In the Proceedings of the International Conference RANLP-2009. Pages 118-122. 2009.