International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 02, 2020
ISSN: 1475-7192

Preparatory Document Structuring Technique

[1]Yan Puspitarani, [2]Ulil Surtia Zulpratita

Abstract---*The need for mining structured data has increased in the past few years. This structured data is used as input for data mining tasks. Text mining is part of data mining where the data used is in the form of unstructured text. Text mining can able to handle unstructured or semi-structured data sets such as emails HTML files and full text documents etc. The unstructured data usually refers to information that does not reside in a traditional row-column database and it is the opposite of structured data. In order to extract information from text, preprocessing steps are needed.This paper discussed about the theoretical basis of preprocessing document for Text Mining. Brief descriptions of some representative approaches such as NLP tasks and Information extraction are provided as well.*

Keywords---*Text mining, Document structuring, Information extraction.*

I. INTRODUCTION

Text mining and data mining systems show many similarities in high-level architecture. Both of them will find useful information from data sources through identifying and exploring interesting patterns. The difference lies on the preprocessing. Data mining assumes that data has been stored in a structured format, most of the preprocessing focus falls on two important tasks: Rubbing and normalizing data and creating a large number of combined tables. In contrast, for text mining systems, preprocessing operations are centered on identifying and extracting representative features for natural language documents. This preprocessing operation is responsible for converting unstructured data stored in a document collection into a more explicit structured intermediary format [1]. This is not done in data mining. In addition, text mining also refers to advances in other computer science disciplines related to handling natural language as its main center. So that, preprocessing is the main task in text mining system.

Many studies have examined various techniques for text preprocessing to make better performance. Kadhim implemented TF-IDF and singular value decomposition (SVD) dimensionality reduction techniques. The proposed system presented an effective preprocessing and dimensionality reduction techniques which help the document clustering by using k-means algorithm. Finally, the experimental results showed that the proposed method enhanced the performance of English text document clustering. Simulation results on BBC news and BBC sport datasets showed the superiority of the proposed algorithm [2].

Preprocessing framework for text mining application was designed by gathering the frequently used preprocessing steps. The framework focused on three major preprocessing tasks Expansion, Removal and Tokenization (ERT). The ERT takes corpus as input and generate the list of tokens; the tokens has been equipped to perform all learning algorithms. ERT framework helps to perform all the preprocessing steps quickly and correctly. [3]

Stemming algorithm called the Enhanced Porter's Stemming Algorithm (EPSA) is proposed to make efficient information on Information Retrieval Technique. The objective of this technique is to overcome the drawbacks of the Porter algorithm and improve web searching. The EPSA was applied to two datasets to measure its performance. The result showed improvement of precision over the original Porter algorithm while realizing approximately the same recall percentages. [4]

Moreover, feature selection took the important role to reduce the dimension of text. Ghareb proposes hybrid feature selection approaches based on the Genetic Algorithm (GA) has been proposed to handle the high dimensionality of

[1]*Engineering Department Widyatama University Bandung, Indonesia.*
[2]*Engineering Department Widyatama University Bandung, Indonesia.*
yan.puspitarani@widyatama.ac.id

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020

text feature and improve categorization performance simultaneously. The results showed that these hybrid approaches were more effective than single filter methods for dimensionality reduction because they were able to produce a higher reduction rate without loss of categorization precision in most situations. [5]

In terms of implementation, Sheng presented Juicer, a system for extracting information from email that was serving over a billion Gmail users daily. the design of the system was informed by three key principles: scaling to a planet-wide email service, isolating the complexity to provide a simple experience for the developer, and safeguarding the privacy of users. The research was using case studies of three extraction tasks implemented on this platform—bill reminders, commercial offers, and hotel reservations—to illustrate the effectiveness of the platform despite challenges unique to each task. [6]

The remaining portion of the paper is organized as follows. Section 2 gives the general architecture of text mining system. Section 3 describes document structuring technique. Information Extraction technique are described in Section 4. Conclusion is given in Section 5.

II. GENERAL ARCHITECTURE OF TEXT MINING SYSTEM

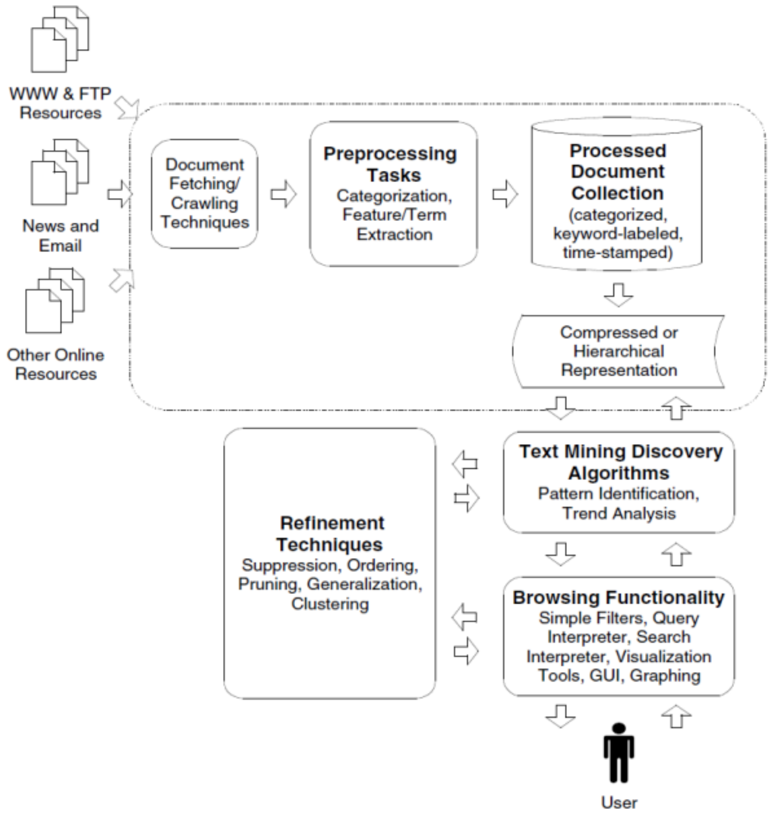

WWW & FTP Resources
News and Email
Other Online Resources
Document Fetching/ Crawling Techniques
Preprocessing Tasks
Categorization, Feature/Term Extraction
Processed Document Collection
(categorized, keyword-labeled, time-stamped)
Compressed or Hierarchical Representation
Text Mining Discovery Algorithms
Pattern Identification, Trend Analysis
Refinement Techniques
Suppression, Ordering, Pruning, Generalization, Clustering
Browsing Functionality
Simple Filters, Query Interpreter, Search Interpreter, Visualization Tools, GUI, Graphing
User

Figure 1:*System architecture for generic text mining system*

A. Preprocessing Tasks

All tasks like routines, processes, and methods required to prepare data for a text mining system's core knowledge discovery operations. These tasks typically center on data source preprocessing and categorization activities. Preprocessing tasks generally convert the information from each original data source into a canonical format before applying various types of feature extraction methods against these documents to create a new collection of documents fully represented by concepts. Where possible, preprocessing tasks may also either extract or apply rules for creating document date stamps, or do both. Occasionally, preprocessing tasks may even include specially designed methods used in the initial fetching of appropriate "raw" data from disparate original data sources [1].

B. Core Mining Operations

This operations are the heart of a text mining system and include pattern discovery, trend analysis, and incremental knowledge discovery algorithms. Among the commonly used patterns for knowledge discovery in textual data are distributions (and proportions), frequent and near frequent concept sets, and associations. Core mining operations can also concern themselves with comparisons between – and the identification of levels of "interestingness" in – some of these patterns. Advanced or domain-oriented text mining systems, or both, can also augment the quality of their various operations by leveraging background knowledge sources. These core mining operations in a text mining system have also been referred to, collectively, as *knowledge distillation* processes [1].

C. Presentation Layer Components

Presentation Layer included GUI and pattern browsing functionality as well as access to the query language. Visualization tools and user-facing query editors and optimizers also fall under this architectural category. Presentation layer components may include character-based or graphical tools for creating or modifying concept clusters as well as for creating annotated profiles for specific concepts or patterns [1].

D. Refinement Technique

The last process included methods that filter redundant information and cluster closely related data but may grow, in a given text mining system, to represent a full, comprehensive suite of suppression, ordering, pruning, generalization, and clustering approaches aimed at discovery optimization. These techniques have also been described as *postprocessing*[1].

II. DOCUMENT STRUCTURING

Most text documents have an unstructured form. Preprocessing techniques are required to prepare raw unstructured data for text mining than those traditionally encountered in knowledge discovery operations aimed at preparing structured data sources for classic data mining operations.

Two clear ways of categorizing the totality of preparatory document structuring techniques are *according to their task* and *according to the algorithms and formal frameworks that they use*[1].

Task-oriented preprocessing approaches envision the process of creating a structured document representation in terms of tasks and subtasks and usually involve some sort of preparatory goal or problem that needs to be solved such as extracting titles and authors from a PDF document. Other preprocessing approaches rely on techniques that derive from formal methods for analyzing complex phenomena that can be also applied to natural language texts. Such approaches include classification schemes, probabilistic models, and rule-based systems approaches.

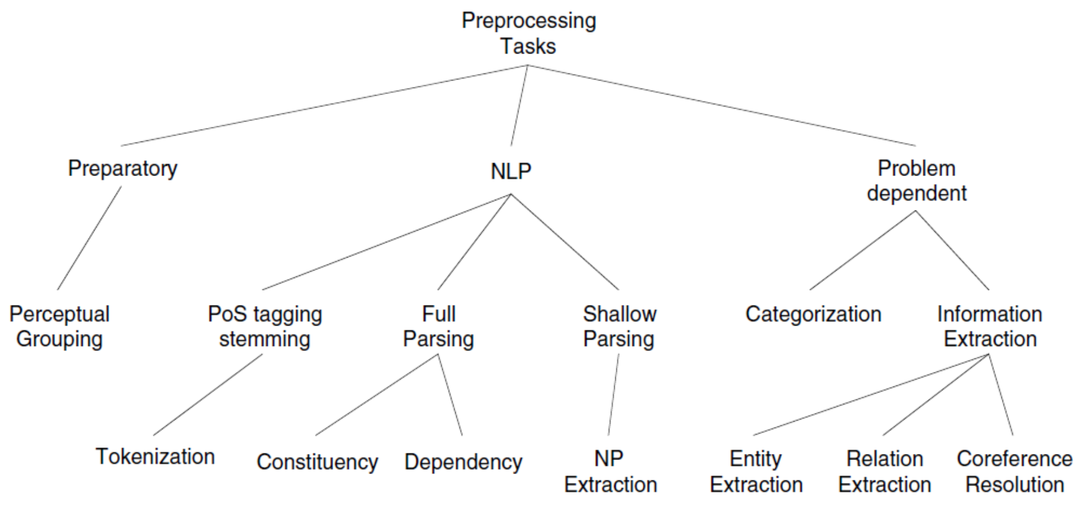


Figure 2: A *taxonomy of text preprocessing tasks*

A divide-and-conquer strategy is typically employed to cope with this extremely difficult problem. The problem is separated into a set of smaller subtasks, each of which is solved separately. The subtasks can be divided broadly into three classes. There are **preparatory processing, general purpose NLP tasks, and problem-dependent tasks**[1].

A. Preparatory Processing

Preparatory processing converts the raw representation into a structure suitable for further linguistic processing. For example, the raw input may be a PDF document, a scanned page, or even recorded speech. The task of the preparatory processing is to convert the raw input into a stream of text, possibly labeling the internal text zones such as paragraphs,

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020

columns, or tables. It is sometimes also possible for the preparatory processing to extract some document-level fields such as <*Author*>or <*Title*>in cases in which the visual position of the fields allows their identification.

B. Natural Language Processing

The general purpose NLP tasks process text documents using the general knowledge about natural language. The tasks may include tokenization, morphological analysis, POS tagging, and syntactic parsing – either shallow or deep. The tasks are general purpose in the sense that their output is not specific for any particular problem. The output can rarely be relevant for the end user and is typically employed for further problem-dependent processing. The domain-related knowledge, however, can often enhance the performance of the general purpose NLP tasks and is often used at different levels of processing.

The NLP components built in this way are valued for their generality. The tasks they are able to perform include tokenization and zoning, part-of-speech tagging and stemming, and shallow and deep syntactic parsing.

1) Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens*, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: Friends, Romans, Countrymen, lend me your ears;

Output: Friends | Romans | Countrymen | lend | me | your | ears

These tokens are often loosely referred to as terms or words, but it is sometimes important to make a type/token distinction. A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A *type* is the class of all tokens containing the same character sequence. A *term* is a (perhaps normalized) type that is included in the IR system's dictionary. The set of index terms could be entirely distinct from the tokens, for instance, they could be semantic identifiers in a taxonomy, but in practice in modern IR systems they are strongly related to the tokens in the document [7].

2) POS tagging

POS tagging is the annotation of words with the appropriate POS tags based on the context in which they appear. POS tags divide words into categories based on the role they play in the sentence in which they appear. POS tags provide information about the semantic content of a word. Nouns usually denote "tangible and intangible things," whereas prepositions express relationships between "things." Most POS tag sets make use of the same basic categories. The most common set of tags contains seven different tags (Article, Noun, Verb, Adjective, Preposition, Number, and Proper Noun). Some systems contain a much more elaborate set of tags. For example, the complete Brown Corpus tag set has no less than 87 basic tags. Usually, POS taggers at some stage of their processing perform morphological analysis of words. Thus, an additional output of a POS tagger is a sequence of stems (also known as "lemmas") of the input words [1].

3) Stemming and Lemmatization

For grammatical reasons, documents are going to use different forms of a word, such as *organize*, *organizes*, and *organizing*. Additionally, there are families of derivationally relatedwordswith similarmeanings, such as *democracy*, *democratic*, and *democratization*. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. [7]

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the re moval of derivational affixes. *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*. [7]

During the learning process Mustafaraj [8] performed a set of experiments, where: a) no stem information at all was used and b) all words had stem information (tagger + manually created list of stems). These results show approximately 1% improvement in recall and precision when stems instead of original words are used. So that at least for the learning task of annotating text with knowledge roles stem information is not necessarily important, but this could also be due to the fact that a large number of other features besides words are used for learning.

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020

4) Syntactical Parsing

Syntactical parsing components perform a full syntactical analysis of sentences according to a certain grammar theory. Efficient, accurate parsing of unrestricted text is not within the reach of current techniques. Standard algorithms are too expensive for use on very large corpora and are not robust enough. Shallow parsing compromises speed and robustness of processing by sacrificing depth of analysis. [1]

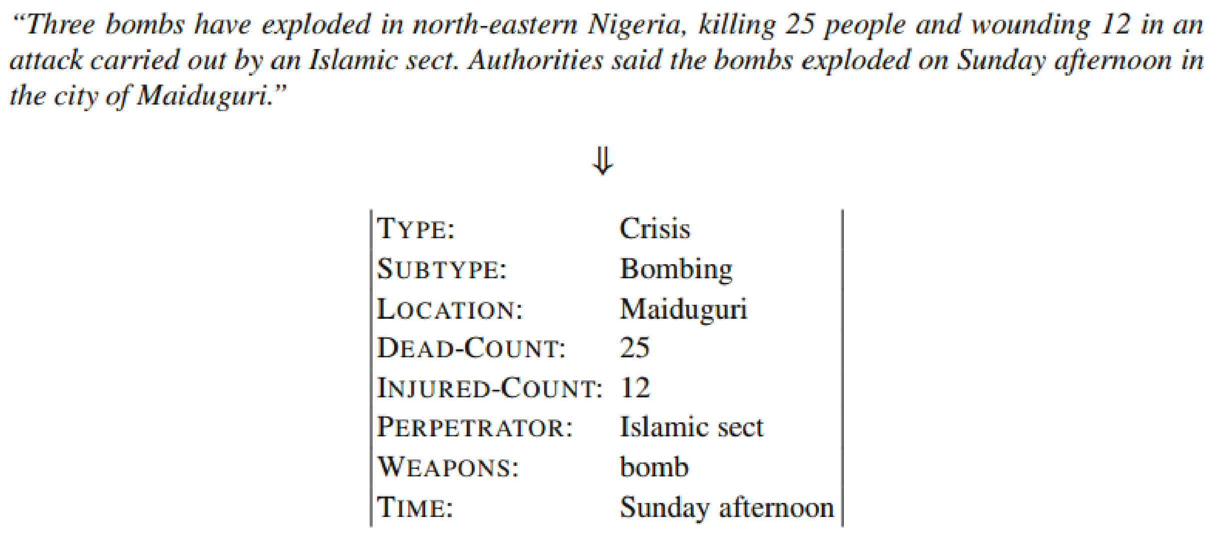
Syntactical parsing is one of the most important steps in the learning framework, since the produced parse trees serve as input for the creation of features used for learning [8]. There are many parser can be used. Mustafaraj [8] experimented with three different parsers: the Stanford parser, the BitPar parser, and the Sleepy parser. All the tested parsers make errors during parsing. In the end, based on speed and output information. Sleepy was the fastest and had the most informative output (it prints the log value expressing the likelihood of parsing, and it labels the majority of nodes with their grammatical function).

C. Problem dependent

The problem-dependent tasks prepare the final representation of the document meaning. In text mining, *categorization* and *information extraction* are typically used.

III. INFORMATION EXTRACTION

The task of Information Extraction (IE) is to identify a predefined set of concepts in a specific domain, ignoring other irrelevant information, where a domain consists of a corpus of texts together with a clearly specified information need. In other words, IE is about deriving structured factual information from unstructured text. [9]



"Three bombs have exploded in north-eastern Nigeria, killing 25 people and wounding 12 in an attack carried out by an Islamic sect. Authorities said the bombs exploded on Sunday afternoon in the city of Maiduguri."

⇓

TYPE:	Crisis
SUBTYPE:	Bombing
LOCATION:	Maiduguri
DEAD-COUNT:	25
INJURED-COUNT:	12
PERPETRATOR:	Islamic sect
WEAPONS:	bomb
TIME:	Sunday afternoon

Figure 3:*Example of automatically extracted information from a news article on a terrorist attack*

Figure 3 shows an example of a text snippet from a news article about a terrorist attack and a structured information derived from that snippet. The process of extracting such structured information involves identification of certain small-scale structures like noun phrases denoting a person or a person group, geographical references and numeral expressions, and finding semantic relations between them. However, in this scenario some domainspecific knowledge is required (understanding the fact that terrorist attacks might result in people being killed or injured) in order to correctly aggregate the partially extracted information into a structured form. [9]

The classic IE tasks include:

1) **Named Entity Recognition (NER)** addresses the problem of the identification (detection) and classification of predefined types of named entities, such as organizations (e.g., 'World Health Organisation'), persons (e.g., 'Muammar Kaddafi'), place names (e.g., 'the Baltic Sea'), temporal expressions (e.g., '1 September 2011'), numerical and currency expressions (e.g., '20 Million Euros'), etc.
2) **Co-reference Resolution (CO)** requires the identification of multiple (coreferring) mentions of the same entity in the text. Entity mentions can be named, pronominal, nominal, and implicit.

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020

3) **Relation Extraction (RE)** is the task of detecting and classifying predefined relationships between entities identified in text.
4) **Event Extraction (EE)** refers to the task of identifying events in free text and deriving detailed and structured information about them, ideally identifying who did what to whom, when, where, through what methods (instruments), and why.

A. The Architecture of IE

There are certain core components shared by most IE systems. The overall IE processing chain can be analyzed along several dimensions. The chain typically includes domain-independent vs. domain-specific components. [9]

A typical architecture of an IE system is depicted in Figure 4 The domain-independent part usually consists of language-specific components that perform linguistic analysis in order to extract as much linguistic structure as possible. Usually, the following steps are performed:

- Meta-data analysis: Extraction of the title, body, structure of the body (identifi- cation of paragraphs), and the date of the document.
- Tokenization: Segmentation of the text into word-like units, called tokens and classification of their type, e.g., identification of capitalized words, words written in lowercase letters, hyphenated words, punctuation signs, numbers, etc.
- Morphological analysis: Extraction of morphological information from tokens which constitute potential word forms—the base form (or lemma), part of speech, other morphological tags depending on the part of speech: e.g., verbs have features such as tense, mood, aspect, person, etc. Words which are ambiguous with respect to certain morphological categories may undergo disambiguation. Typically part-of-speech disambiguation is performed.
- Sentence/Utterance boundary detection: Segmentation of text into a sequence of sentences or utterances, each of which is represented as a sequence of lexical items together with their features.
- Common Named-entity extraction: Detection of domain-independent named entities, such as temporal expressions, numbers and currency, geographical references, etc.
- Phrase recognition: Recognition of small-scale, local structures such as noun phrases, verb groups, prepositional phrases, acronyms, and abbreviations.
- Syntactic analysis: Computation of a dependency structure (parse tree) of the sentence based on the sequence of lexical items and small-scale structures. Syntactic analysis may be deep or shallow. In the former case, one is interested in computing all possible interpretations (parse trees) and grammatical relations within the sentence. In the latter case, the analysis is restricted to identification of non-recursive structures or structures with limited amount of structural recursion, which can be identified with a high degree of certainty, and linguistic phenomena which cause problems (ambiguities) are not handled and represented with underspecified structures.

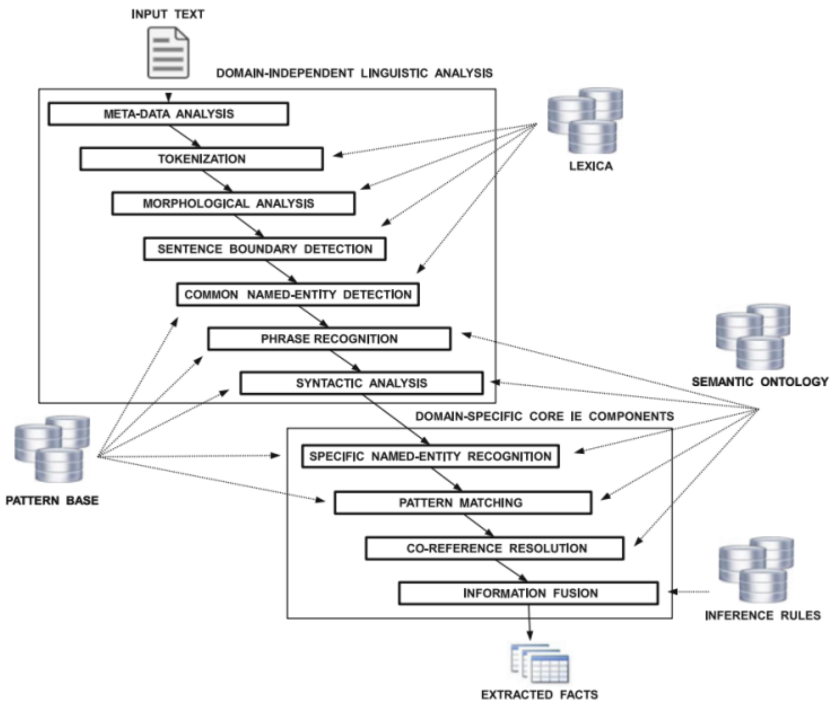


Figure 4:*Typical architecture of an information extraction system*

In the domains specific core of the processing chain, a NER component is applied to identify the entities relevant in a given domain. Patterns may then be applied to: (a) identify text fragments, which describe the target relations and events, and (b) extract the key attributes to fill the slot in the template representing the relation/event. A co-reference component identifies mentions that refer to the same entity. Finally, partially-filled templates are fused and validated using domain-specific inference rules in order to create full-fledged relation/event descriptions. The last step is crucial since the relevant information might be scattered over different sentences or even documents. It is important to note that, in practice, the borders between domain-independent linguistic analysis components and core IE components may be blurred, e.g., there may be a single NER component which performs domain independent and domain-specific named-entity extraction simultaneously.

B. Comprehensive Stage-by-Stage Example

***Original Sentence*:** Mr. Eskew was Vice President of Worldwide Sales for Sandpiper Networks, which was recently acquired by Digital Island where he created the worldwide sales strategy.

***After Part of Speech Tagging*:**

<Prop>Mr. Eskew</Prop><Verb>was</Verb><Prop>Vice President</Prop><Prep>of</Prep><Prop>Worldwide Sales</Prop><Prep>for</Prep><Prop>Sandpiper Networks</Prop>which<Verb>was</Verb><Adv>recently</Adv><Verb>acquired</Verb><Prep>by</Prep>< Prop>Digital Island</Prop>where <Pron>he</Pron><Verb>created</Verb><Det>the</Det><Adj>worldwide</Adj><Nn>sales strategy.</Nn>

***After Shallow Parsing*:**

NP:*{*Mr. Eskew*}* was NP:*{*Vice President ofWorldwide Sales*}* for NP:*{*Sandpiper Networks*}* which was ADV:*{*recently*}* V:*{*acquired*}* by NP:*{*Digital Island*}* where NP:*{*he*}* V:*{*created*}* NP:*{*the worldwide sales strategy*}*

***After Named Entity Recognition*:**

Person:*{*Mr. Eskew*}* was Position:*{*Vice President of Worldwide Sales*}* for Company:*{*Sandpiper Networks*}* which was ADV:*{*recently*}* V:*{*acquired*}* by Company:*{*Digital Island*}* where Person:*{*he*}* V:*{*created*}* NP:*{*the worldwide sales strategy*}*

***After Merging (Anaphora Resolution)*:**

Person:*{*Mr. Eskew*}* was Position:*{*Vice President of Worldwide Sales*}* for Company:*{*Sandpiper Networks*}* which was ADV:*{*recently*}* V:*{*acquired*}* by Company:*{*Digital Island*}* where Person:*{*Mr. Eskew*}* V:*{*created*}* NP:*{*the worldwide sales strategy*}*

***Frames Extracted*:**

Frame Type: **Acquisition**

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020

Acquiring Company: Digital Island
Acquired Company: Sandpiper Networks
Acquisition Status: Historic
FrameType: **PersonPositionCompany**
Person: Mr. Eskew
Position: Vice President of Worldwide Sales
Company: Sandpiper Networks
Status: Past

C. NER Approach

NER is a fundamental task and it is the core of natural language processing (NLP) system. NER involves two tasks, which is firstly the identification of proper names in text, and secondly the classification of these names into a set of predefined categories of interest, such as person names, organizations (companies, government organisations, committees, etc), locations (cities, countries, rivers, etc), date and time expressions. [10] There are three standard approach to NER :

1) Hand-written regular expressions

Regular expression (REGEX) spotting consists in detecting patterns sequence of characters that obey certain rules described using meta models such as lower cases (#[a-z]#), upper cases (#[A-Z]#) or Digits (#[0- 9]#). Detecting such regular expression in handwritten documents can be useful for finding sub-string which are relevant for a further higher level information extraction task. For example, a system of that kind could spot entities. For example, spotting date (#[0-9]{2}/[0-9]{2}/[0-9]{4}#), first name (#[A-Z][a-z]*#), ZIP code and city name of a french postal address (#[0-9]{5} [A-Z]*#). The extraction of these informations allow to consider high level processing stages such as document categorisation, customer identification, Named Entity detection, etc [11]

2) Using classifiers

Consider a classification problem in which we want to learn to distinguish between elephants (y = 1) and dogs (y = 0), based on some features of an animal. Given a training set, an algorithm like logistic regression or the perceptron algorithm (basically) tries to find a straight line—that is, a decision boundary—that separates the elephants and dogs. Then, to classify a new animal as either an elephant or a dog, it checks on which side of the decision boundary it falls, and makes its prediction accordingly.

Here’s a different approach. First, looking at elephants, we can build a model of what elephants look like. Then, looking at dogs, we can build a separate model of what dogs look like. Finally, to classify a new animal, we can match the new animal against the elephant model, and match it against the dog model, to see whether the new animal looks more like the elephants or more like the dogs we had seen in the training set.

Algorithms that try to learn $p(y|x)$ directly (such as logistic regression), or algorithms that try to learn mappings directly from the space of inputs X to the labels {0, 1}, (such as the perceptron algorithm) are called discriminative learning algorithms. Here, we’ll talk about algorithms that instead try to model $p(x|y)$ (and $p(y)$). These algorithms are called generative learning algorithms. For instance, if y indicates whether an example is a dog (0) or an elephant (1), then $p(x|y = 0)$ models the distribution of dogs’ features, and $p(x|y = 1)$ models the distribution of elephants’ features. [12]

3) *Sequence models*

Named Entity Recognition (NER) is a task that seeks to locate and classify entities (‘atomic elements’) in a text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, etc. It can be viewed as a two stage process: 1. Identification of entity boundaries 2. Classification into the correct category For example, if “Mahatma Gandhi” is a named entity in the corpus, it is necessary to identify the beginning and the end of this entity in the sentence. Following this step, the entity must be classified into the predefined category, which is NEP (Named Entity Person) in this case. [13]

In Machine Learning-based NER system, the purpose of Named Entity Recognition approach is converting identification problem into a classification problem and employs a classification statistical model to solve it. In this type of approach, the systems look for patterns and relationships into text to make a model using statistical models and machine learning algorithms. The systems identify and classify nouns into particular classes such as persons, locations, times, etc base on this model, using machine learning algorithms. There are two types of machine learning model that are use for NER. Supervised and Unsupervised machine learning model. Supervised learning involves using a program that can learn to classify a given set of labeled examples that are made up of the same number of features. Each example is thus represented with respect to the different feature spaces. The learning process is called supervised, because the people who marked up the training examples are teaching the program the right distinctions. The supervised learning approach requires preparing labeled training data to construct a statistical model, but it cannot achieve a good performance without a large amount of training data, because of data sparseness problem. [10]

Unsupervised learning method is another type of machine learning model, where an unsupervised model learns without any feedback. In unsupervised learning, the goal of the program is to build representations from data. These representations can then be used for data compression, classifying, decision making, and other purposes. Unsupervised learning is not a very popular approach for NER and the systems that do use unsupervised learning are usually not completely unsupervised. [10]

Machine-learning (ML) approach Learn rules from annotated corpora. Machine learning approach is commonly used for NER because training is easy, same ML system can be used for different domains and languages and their maintenance is also less expensive. There are various machine learning approaches for NER such as CRF (conditional Random Fields), MEMM (Maximum Entropy Markov Model), SVM (Support Vector Machine) and HMM (Hidden Markov Model) and dictionary based approach. Among all these HMM, being the most promising, has not been explored in its full potential for NER. The work that has been reported is domain specific and does not establish it as a general technique. [14]

D. Evaluation in Information Extraction

Given an input text, or a collection of texts, the expected output of an IE system can be defined very precisely. This facilitates the evaluation of different IE systems and approaches. In particular, the precision and recall metrics were adopted from the IR research community for that purpose. They measure the system's effectiveness from the user's perspective, i.e., the extent to which the system produces all the appropriate output (recall) and only the appropriate output (precision). Thus, recall and precision can bee seen as measure of completeness and correctness, respectively.

To define them formally, let #key denote the total number of slots expected to be filled according an annotated reference corpus, representing ground truth or a "gold-standard", and let #correct (#incorrect) be the number of correctly (incorrectly) filled slots in the system's response. A slot is said to be filled incorrectly either if it does not align with a slot in the gold standard (spurious slot) or if it has been assigned an invalid value. Then, precision and recall may be defined as follows:

$$precision = \frac{\#correct}{\#correct + \#incorrect} \quad (1)$$

$$recall = \frac{\#correct}{\#key} \quad (2)$$

In order to obtain a more fine-grained picture of the performance of IE systems, precision and recall are often measured for each slot type separately. The f-measure is used as a weighted harmonic mean of precision and recall, which is defined as follows:

$$F = \frac{(\beta^2+1) \times precision \times recall}{(\beta^2 \times precision) + recall} \quad (3)$$

In the above definition β is a non-negative value, used to adjust their relative weighting (β=1.0 gives equal weighting to recall and precision, and lower values of β give increasing weight to precision). Other metrics are used in the literature as well, e.g., the so called slot error rate, SER [15], which is defined as follows:

$$SER = \frac{\#incorrect + \#missing}{\#key} \quad (4)$$

where #missing denotes the number of slots in the reference that do not align with any slots in the system response. It reflects the ratio between the total number of slot errors and the total number of slots in the reference. Depending on the particular needs, certain error types, (e.g., spurious slots) may be weighted in order to deem them more or less important than others.

IV. CONCLUSION

This paper has been given information about preprocessing textual data to be structured based on NLP and Information Extraction. From this explanation it can be seen that document structuring is not easy and requires expensive computing. There are a lot of complicated processes are passed before the document is structured.

This approach still requires a long research for an expensive effort. However, the current needs of users wanting to extract information are as natural as possible. the latest technology is needed to make computer analysis capabilities resemble human analytical capabilities.

REFERENCES

1. Ronen Feldman and James Sanger, *The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data*. Cambridge, UK: Cambridge University Press, 2007.
2. Ammar Ismael Kadhim, Yu-N Cheah, and Nurul Hashimah Ahamed, "Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering," in *Artificial Intelligence with Applications in Engineering and Technology*, 2014, pp. 69-73.
3. E. Elakiya and N. Rajkumar, "Designing preprocessing framework (ERT) for text mining application," in *2017 International Conference on IoT and Application (ICIOT)*, Nagapattinam, 2017.
4. Safaa I. Hajeer, Rasha M. Ismail, Nagwa L. Badr, and Mohamed Fahmy Tolba, "A New Stemming Algorithm for Efficient," *Multimedia Forensics and Security*, pp. 117-135, 2017.
5. Abdullah Saeed Ghareb, Azuraliza Abu Bakar, and Abdul Razak Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Systems with Applications*, pp. 31-47, May 2016.
6. Ying Sheng, Sandeep Tata, and James B. Wendt, "Anatomy of a Privacy-Safe Large-Scale Information Extraction System Over Email," in *KDD*, London, 2018, pp. 734-743.
7. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *An Introduction of Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.
8. Eni Mustafaraj, Martin Hoof, and Bernd Freisleben, *Mining Diagnostic Text Reports by Learning to Annotate Knowledge Roles*, Anne Kao and Stephen R. Poteet, Eds. Washington, United State of America: Springer, 2017.
9. Jakub Piskorski and Roman Yangarber, "Information Extraction: Past, Present and Future ," in *Theory and Applications of Natural Language Processing*, T. Poibeau, Ed. Berlin: Springer, 2013, ch. 2, p. 23.
10. Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat, "Named Entity Recognition Approaches," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, February 2008.
11. Gautier Bideaulta, Luc Mioulet, Cl´ement Chatelain, and Thierry Paquet, "Spotting Handwritten Words and REGEX using a two stage BLSTM-HMM architecture," *Document Recognition and Retrieval*, 2015.
12. Andrew Ng, *Generative Learning Algorithms*., ch. 4.
13. Karthik Gali, Harshit Surana, Ashwini Vaidya, Praneeth Shishtla, and Dipti Misra Sharma, "Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition," in *IJCNLP-08 Workshop on NER for South and South East Asian Languages*, Hyderabad, 2008, pp. 25-32.
14. Sudha Morwal, Nusrat Jahan, and Deepti Chopra, "Named Entity Recognition using Hidden Markov Model (HMM)," *International Journal on Natural Language Computing (IJNLC)*, vol. 1, pp. 15-23, December 2012.
15. J. Makhoul, F.Kubala, R.Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proceedings of DARPA Broadcast News Workshop*, Herndon, 1999.
16. Sisyagina, E.P., Pashkina, Y.V., Molev, A.I., Gorchakova, N.G., Samodelkin, A.G., Sisyagin, P.N., Sochnev, V.V., Kozyrenko, O.V. Correcting the immune status of the calves' organisms on the background of their associative respiratory diseases(2018) International Journal of Pharmaceutical Research, 10 (4), pp. 749-754. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061870927&doi=10.31838%2fijpr%2f2018.10.04.130&partnerID=40&md5=5ee42576e88026f7677f7c5852f8b974
17. Bhausaheb b. Jankar, devesh d. Gosavi (2017) adverse drug reaction of lithium carbonate-a review. Journal of Critical Reviews, 4 (1), 1-6. doi:10.22159/jcr.2017v4i1.14555
18. Sabale V, Sakarkar SN, Pund S, Sabale PM. "Formulation and Evaluation of Floating Dosage Forms: An Overview." Systematic Reviews in Pharmacy 1.1 (2010), 33-39. Print. doi:10.4103/0975-8453.59510

DOI: 10.37200/IJPR/V24I2/PR200640
Received: 02 Jan 2020 | Revised: 15 Jan 2020 | Accepted: 28 Jan 2020