Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Information extraction and representation from free text reports

Isha Saxena

Orientador(es) | Paulo Miguel Quaresma
Teresa Gonçalves

Évora 2021

Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Information extraction and representation from free text reports

Isha Saxena

Orientador(es) | Paulo Miguel Quaresma
Teresa Gonçalves

Évora 2021

A dissertação foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente	Irene Pimenta Rodrigues (Universidade de Évora)
Vogais	Teresa Gonçalves (Universidade de Évora) (Orientador) Vitor Beires Nogueira (Universidade de Évora) (Arguente)

Évora 2021

Acknowledgement

I feel immense pleasure in expressing my profound sense of reverence and gratitude to my thesis supervisors Dr. Paulo Quaresma and Dr. Teresa Gonçalves, for their valuable and meticulous guidance, motivation, constructive criticism, constant inspiration and supervision throughout the course of study.

I would also like to express my profound gratitude to my parents for providing me with unfailing support and continuous encouragement. This accomplishment would not have been possible without them.

Lastly, I must express my gratitude to my friends and colleagues for helping me and supporting me throughout our years of study.

Isha Saxena
November, 2020

Contents

List of Figures

List of Tables

Acronyms

ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary verb
CCONJ	coordinating conjunction
CE	Concept Extraction
CRF	Conditional Random Field
DARPA	Defense Advanced Research Project Agency
DET	determiner
ECT	Escola de Ciências e Tecnologia
HMM	Hidden Markov Model
IE	Information Extraction
IR	Information Retrieval
MUC	Message Understanding Conference
NASA	National Aeronautics and Space Administration
NER	Named Entity Recognition
OP	optional
PART	particle
POS	Parts of Speech
SE	Systems Engineer
SEVA	Systems Engineers Virtual Assistant
SVM	Support Vector Machine
UE	Universidade de Évora

Abstract

The need for extracting specific information has increased drastically with the boost in digital-born documents. These documents majorly comprise of free text from which structured information can be extracted. The sources include, customer review reports, patient records, financial and legal documents, etc. The needs and applications for extracting specific information from free text are growing every moment, and new researches are emerging to mine contextual information in a way that is both highly efficient and convenient in its usage.

This thesis work address to the problem of extracting specific information from free text, specifically for the domains who lack labeled data. First step in the development of an advanced information extraction system is to extract and represent structured information from unstructured natural language text. To accomplish this task, the thesis proposes a system for extracting and tagging domain specific information, as domain related entities / concepts, and relational phrases. The approaches comprise of dictionary matching for domain specific concept extraction, and rule based pattern matching for relation extraction and tagging the free text accordingly. The experiments were performed on Altice Labs'[1] customer reports. The system achieved over 80% recall and 90% precision for both concept and relation extraction.

The proposed domain-specific concept extraction module was compared with existing concept extraction platforms: Microsoft Concept Graph[2] and DBpedia Spotlight[3]. The proposed model yielded high performance results then both the platforms.

Keywords: Information Extraction, Concept Extraction, Relation Extraction, Dictionary Matching, Rule Based Approach, Free Text Tagging

[1] https://www.alticelabs.com/en/

[2] https://concept.research.microsoft.com/

[3] https://www.dbpedia-spotlight.org/

Sumário

Extração e representação de informações de relatórios de texto livre

A necessidade de extrair informações específicas aumentou drasticamente com o aumento dos documentos de origem digital. Esses documentos consistem principalmente de texto livre do qual informações estruturadas podem ser extraídas. As fontes incluem relatórios de revisão de clientes, registos de pacientes, documentos financeiros e jurídicos, etc. As necessidades e aplicações para extrair informações específicas de texto livre estão crescendo a cada momento e novas pesquisas estão surgindo para extrair informações contextuais de uma forma altamente eficiente e conveniente em seu uso.

Este trabalho aborda o problema da extração de informações específicas em texto livre, especificamente para os domínios que carecem de dados etiquetados. O primeiro passo no desenvolvimento de um sistema avançado de extração de informações é extrair e representar informações estruturadas de um texto de linguagem natural não estruturado. Para cumprir essa tarefa, a tese propõe um sistema para extrair e marcar informações específicas do domínio, como entidades / conceitos relacionados ao domínio e frases relacionais. As abordagens incluem correspondência de dicionário para extração de conceitos específico de domínio e correspondência de padrão baseada em regras para extração de relação e marcação de texto livre. As experiências foram realizados nos relatórios de clientes [4] da Altice Labs. O sistema atingiu mais de 80 % de recall e 90% de precisão para extração de conceito e relação.

O módulo de extração de conceito específico de domínio proposto foi comparado com plataformas de extração de conceito existentes: Microsoft Concept Graph [5] e DBpedia Spotlight [6]. O modelo proposto rendeu resultados de alto desempenho para ambas as plataformas.

[4] https://www.alticelabs.com/en/

[5] https://concept.research.microsoft.com/

[6] https: //www.dbpedia-spotlight .org /

Palavras chave: Extração de Informação, Extração de Conceito, Extração de relação, Correspondência de dicionário, Abordagem Baseada em Regras, Etiquetagem de texto livre

Chapter 1

Introduction

The word '*inform*' is derived from the Latin word '*informare*' which means '*to give, educate or instruct*', and the word '*information*' is derived from the Latin word '*informacion*' (old french '*enformacion*') which means '*to give form to mind*'. Information is, gathering knowledge on any specific or general subject by commutation, study, research, etc.

Information helps living beings in understanding and analyzing their surroundings. For example, in the sentence: *Pingo Doce will operate from 8:00 to 19:00 during COVID-19*, humans can easily extract the information that a supermarket named Pingo Doce will be open from the time of 8:00 in the morning till 19:00 in the evening during the outbreak of the COVID-19 disease. But, for a computer, Pingo Doce and COVID-19 are just some words and, 8:00 and 19:00 are some numerical values. To make a computer able to extract this kind of information from plain text, Information Extraction (IE) techniques have emerged. A very neat example of IE is the system that extracts the date and time from email messages and marks them as a reminder in a calendar application.

In this era of digitalization, information extraction plays a vital role in extracting knowledge from the billions of data, that are generated every day from all the domains such as medical, news, legal, customer reviews, etc. The data from different domain posses different categories of information that needs to be extracted. For this reason, IE is not fully domain independent, and extracting specific information from all available data is practically impossible for humans. But, by using IE techniques it can be done with high precision and much less effort. The aim of this thesis work is to develop a system for extracting domain-specific information from plain text data.

1.1 Motivation

The Altice Labs[1] project '*SIGO*' has been a primary motivation for this thesis work. Altice Labs is an organization that focuses on research and development of telecommunication technologies. It was established in 1950, and since then it is placing benchmarks for innovation and future technologies. The project '*SIGO*' was a collaborated project between Altice Labs and University of Évora. Being a leading provider of products and services in telecommunication industry, Altice, Altice Labs' mother company, receives a huge amount of customer reports every day. The project aims at developing an efficient system for extracting IT related entities from the queries.

The knowledge acquired from working on this real-life project inspired and motivated me to work in the field of IE and NLP. Thus, I decided to develop an efficient system for domain-specific information extraction for the thesis work.

1.2 Goals and Objectives

The main goal of this work is to develop a system that can automatically extract domain-specific information from unstructured natural language text. The following objectives were set to achieve the goal:

- study the related work on domain-specific information extraction,
- propose approaches and techniques to extract domain-specific information,
- apply the proposed approaches and techniques to develop a system for domain-specific information extraction,
- test and evaluate the developed system, and
- compare the developed system with related work.

The second goal of this work is to apply the proposed methodologies and approaches for extracting IT related entities, and relation from Altice Labs' customer reports. Altice Labs' customer reports majorly comprises of queries regarding their system and services. Hence, upon extracting IT specific information from customer queries will ease the task of understanding the problem.

[1] https://www.alticelabs.com/en/index.html

1.3 Proposal and Approaches

This work presents an approach for extracting specific information, namely identification of domain-related concepts and relations from plain text. To reach the goal mentioned in the previous section, the envisaged proposal is composed of three different modules:

- a concept extraction module that uses domain-specific dictionary matching,
- a relation extraction module that uses a rule based approach, and
- combination module that joins the concepts and relations obtained to tag complete sentences using a rule based matching approach.

1.4 Main contributions

The main contributions of this thesis work are:

- a composite survey on the related work of specific information extraction conducted on the datasets of various domains,
- a concept extraction module,
- a relation extraction module, and
- an output tagging module that combines the output of the concept and relation extration modules.

The developed system can be used for extracting specific information from plain text and tagging the data accordingly.

1.5 Thesis Outline

This thesis comprises of five chapters. Chapter 2 describes information extraction in detail including, concept extraction and relation extraction. This chapter also presents the related work done in the respective fields of research. Chapter 3 proposes the system architecture and the tools and techniques used for developing the system. Chapter 4 describes the results obtained when applying the developed system to Altice Labs' data 4.1 along with its evaluation and comparative analysis.

Chapter 5 presents the conclusion of the thesis work along with its limitations and possible future work.

Chapter 2

Information Extraction

2.1 Overview of Information Extraction

Information Extraction (IE) refers to the task of automatically extracting suitable information from textual data sources. With the advancement of multimedia applications, content extraction from audio, video and images sources is also seen as information extraction [1][2]. Information majorly consists of entities, relationship between entities, attributes which describe entities, concepts and terminologies.

The understanding of what is information varies from data taken into consideration and depends on which content is required to be extracted. For example in the sentence '*Larry Page and Sergey Brin are American computer scientists who invented PageRank algorithm*' so the question '*who is Larry Page?*' requires extraction of attribute '*American*' and '*computer scientist*' for the mentioned entity '*Larry Page*'. However, the question '*who invented PageRank Algorithm*' requires extraction of entities '*Larry Page*', '*Sergey Brin*' and extraction of relationship '*invented*'. From this example it can be seen that different information should be extracted from same data content based upon the need.

Information Extraction is often confused with two different fields namely: Information Retrieval (IR) and full text understanding. Full text understanding is a much bigger area in terms of implementation, difficulty and scope. According to [3], "*Information Extraction is a more limited task than full text understanding*". Full text understanding requires a computer to understand the full essences of the text, which is sometimes rather difficult for humans as well. Information extraction on the other hand is rather simple, as it requires only the understanding of specific segments of text, and remover ignores the connecting phrases and stop words. According to researchers, compared to full text understanding, information retrieval and information extraction are both simpler tasks. However, researchers dont possess the same view on the comparison of IR and IE. Information Retrieval is defined as the task of finding relevant documents(usually text), from a collection of documents, that match with user's query [4]. On the contrary, IE extracts predefined features like entities, its attributes, etc. Google is a classic example of IR. Cowie and Wilks [5] states that IR is more advanced and mature then IE. However, [6] states that both of them are difficult but IE is more difficult then IR as it aims to attain detailed information from the documents, like text features and relationships between pieces of text.

From the early 1960's researchers have been working in the field of information extraction[7] [8]. In accounts, the first successful work of information extraction was accomplished in 1970 by Sager and her team. They developed a system to extract medical information from patients' records by using dictionary and pattern matching [8] [9]. In 1987, the Message Understanding Conference(MUC) gave a boost up to the IE field; the Defense Advanced Research Project Agency (DARPA) of U.S. Defense Department organized and financed 7 MUC competition from 1989 to 1997 [1] [7] [8] [9] [10] [2]. In chronological order the goals of these conferences were to extract: information from short naval messages, data on terrorists from newspapers and from joint ventures, space vehicles, and missile launches from news articles [1] [8]. In early 1990's the first successful commercial use of IE was established by the JASPER System [5] [1]. It was build for Reuters, a news agency, to provide real-time financial news. It was developed using a great amount of handcrafted rules without any underlying learning algorithm [5].

Tools and techniques for information extraction have developed rapidly since its advent. The early systems of IE were rule based, developed using heavy hand crafted rules [11] [12]. As the variety of data grew bigger with time, manual coding of rules for all possible scenarios became wearisome. To cope up with this, researchers developed algorithms which were able to automatically learn rules from examples [13] [14]. Additionally, with the increasing access of internet, the amount of unstructured data increased rapidly, and thus rule based approaches for such noisy data became less efficient. To overcome this, statistical learning approaches were adopted. Naïve Bayes classifier, maximum entropy models and Sequence Models like Hidden Markov Model (HMM), Conditional Random Field(CRF) [2] [15] [16] [17] [18] [19] were used [1]. Nowadays, Deep Learning Neural Networks like RNN and CNN, and other Machine Learning methods are used for extracting relevant information efficiently [20] [21] [22]. NLP tools like POS taggers, language models, special Python libraries like SPACY [23] and NLTK [24] and word to vector conversion are used for preprocessing the data.

With the growing amount of unstructured data sources, it has now become essential to extract information in structured forms in order to evaluate it and learn from it. A very vital example of this is Censorship of Twitter, which aims at blocking the users who tweet hate speeches. Twitter is just one social media platform, however there are many others like Facebook, Instagram, etc. where unstructured data is in abundance. It is crucial that it gets monitored, thus automatic extraction of structured information is very essential. Not only this, various enterprises happen to have a lot unstructured data, like customer reviews, which when processed can be used to improve product/service quality. Also, biomedical text data includes various texts like, patient reports and medical history, records of proteins, genes, medications, etc; researchers work on extracting this information from raw data sources to form a knowledge base. With the help of advanced data mining techniques this structured information helps in predicting possible diseases, better understanding of treatment and many more. Comparison shopping, showing advertisement of one's interest by ad sensing, event extraction from news, populating existing knowledge bases and record management are also major applications of Information Extraction.

2.2 Types of Text Data

Textual data is the biggest and most abundant source of data. It usually consists of documents that represent words, sentences or even paragraphs of free flowing text [25]. Text data can contain alphabets, digits and special characters. Based upon its format, text can be divided into three categories, structured, semi-structured and unstructured text.

2.2.1 Structured Text

Structured text is defined as the text in which the role of every string is clearly specified and, is easily predictable from the structure of the document, e.g. tables and database. In structured text, the strings of variable lengths like names, date, currency, ZIP codes, etc. are stored in predefined records. This type of text can be both human or machine generated as long as it maintains its format. Structured text can be easily searched and analyzed with various manually generated queries or machine algorithms. Since, computers can easily understand and search text in database schema, IE research is not relevant as most of the information is already presented. Financial data like fund transfer details, record data like student or customer details and location data are some examples of structured text.

2.2.2 Unstructured Text

All the text that is narrative in nature, without predefined formatting, is unstructured text. Unstructured text cannot be mapped onto standard database fields; however, some fields of database may contain unstructured text. For example, individual user reviews are free/unstructured text in "*Review*" column of a table. The major part of the user generated information is unstructured text, like, emails, blogs, text messages, news articles, journals, reports, etc. [26]. Unstructured data is a vital source of information for business and research, thus, the major focus of IE is to built a system that can extract meaningful information from natural language free text. Significant researches have been made to extract and utilize information from free text, like business organizations often extract information from unstructured text to improve trade intelligence, sentiment analysis from social media data, predictive analysis from patients records, news tracking and many more [27].

Natural language text is often vague and ambiguous, hence many NLP tools and techniques are used to index unstructured text. IE systems are normally composed by rules for free text that are usually based on domain object recognition, semantic categorization and syntactic examination. The rules can be both hand coded or generated from training models. [28] mentions that "*unrestricted natural language understanding is a long way from being solved*"; however, IE provides many useful insights from free text.

2.2.3 Semi-Structured Text

Semi-structured text is the one that lies in between free text and structured text. This kind of text does not necessarily follow a defined format or grammar rules, however it contains metadata and semantic tags to make it more easily recognizable and search able then unstructured data [28]. HTML pages on world wide web are all semi-structured data. HTML pages contain meta tags like keywords, description, type, etc. which helps in indexing the website, but the web page content itself is unstructured. A tag can have multiple parameters, and the number of tags change from page to page. Wrappers like Python wrapper Beautiful Soup [29] are used to extract information from HTML pages.

2.3 Concept Extraction

In general terms, "*Concepts are defined as abstract ideas or general notions that occur in the mind, speech, or thought*" [30]. Concepts are the basic ideas that help human mind to shape and understand the world. Everything which surrounds us is defined by concepts in our mind: from natural things like tress, mountains or people to man made objects like computer and car and even all kinds of emotions like happiness, freedom, etc. are categorized by concepts. Concepts are thoughts stored in our long term memory. In the field of computer science, concepts are the terminology and study of computing paradigm [31]. It includes theories, research methods, hardware / software applications, and many more. Its vastness can be understood by the huge vocabulary of computer science and information technology. The process of finding and extracting these concepts from text is referred to as Concept Extraction.

2.3.1 Named Entity Recognition vs. Concept Extraction

An entity is defined as any person, place, thing or object, regarding which the information is usually gathered[32]. It can be abstract or have a physical existence. Named entities are the entities that are denoted by a proper name. For example '*human*' is an entity however '*Alan Turing*' is a named entity. In NLP, entities are noun phrases that comprises of one or more word tokens, and named entities are majorly proper nouns [33] [34]. Named Entity Recognition (NER) is also known as named entity extraction, or entity identification [35]. It is a sub-task of Information Extraction which involves identifying and classifying named entities from text data.

Concept Extraction is a task similar to Named Entity Recognition, however named entities are general in nature, like, names of persons, organizations or locations and concepts are more domain specific. For instance, if one is interested in extracting specifically IT related concepts from the sentences then, NER does not suffice in this task. For example in the sentence:

"*Hello Altice's Lab, we are noticing a decrease in the available memory percentage, kindly check. Geographic Structure: NEW YORK SHELTER ISLAND. Thanks, James*" we have:

- Named Entities: *Altice's Lab*, *NEW YORK SHELTER ISLAND*, *James*;
- IT Concepts: *available memory percentage.*

For extracting the domain specific concepts, new labels need to be defined. The common factor between named entities and concepts is that they both are majorly recognized as noun phrases.

2.3.2 Tools and Approaches

Many tool and techniques have been introduced to identify and categorize named entities, including concept terms. The approaches include rule based algorithms, supervised, semi-supervised, unsupervised learning methods and neural networks. Rule based approaches include manually built patterns to match sequences of words and to identify single or multiple entities and their boundaries [2] [36] [37]. Supervised learning methods require large amount of labeled data to learn the features, from positive and negative samples, for identifying the given types of entities. Hidden Markov Model (HMM) [38], Maximum entropy Markov models [39], Support Vector Machine (SVM) [40], Conditional Random Field (CRF) [41] and Decision Trees [42] are the main supervised learning methods used. Semi-supervised methods use labeled data to start the entity recognition; then, once the entity is found, it uses the context of entities or the same entity found in different formats to learn new rules [43] [44]. Unsupervised learning methods work on unlabeled data by extracting the semantic and syntactic features from the noun phrases or nouns and then clustering them according to its type [45] [46] [47]. Recently, neural network approaches are explored widely for the NER task. Almost all the neural network approaches require a labeled training set to identify entities and then feed the results into classification models like CRF.

Entities in a NER dataset are annotated at the word-token level using labeling schemes like IOB which gives, the inside (I), outside (O) and beginning (B) representation for each word in an entity [48]. However, to identify domain specific entities/concepts for which labeled data is not available, more low level techniques like dictionary [49] [50] [51] [52], pattern matching, [53] [52] and POS tagging [53] [54] are used.

There are many platforms that are available for domain specific entity extraction. Some of the famous platforms that are discussed in details include, Microsoft Concept Graph, DBpedia Spotlight, Cortical.io. and MonkeyLearn.

Microsoft Concept Graph

Microsoft Concept Graph is a project of Microsoft that aims at understanding the common concepts in natural language [55]. Concepts can be termed as entities which contain facts, and the idea of Microsoft Concept Graph is to map these entities in a format that can enable a machine to conceptualize [56]. The knowledge graph built by the Microsoft research group is called "*Probase*" [57], which is developed by mining knowledge from billions of web pages, blogs and search footprints. It contains about 5.4 million concepts.

The model that maps entities to concepts is called the Conceptualization model. For this research we used the latest release of the Single Instance Conceptualization Model. This model takes a single instance as input and gives its corresponding concept with a probability, as a result. The probability tells how much an instance is close to the concept defined in the graph. There are six ways to get these results for instance *'e'* to be a concept *'c'*, namely:

- P(c|e) (probability of a concept 'c' to an instance 'e'),
- P(e|c) (probability of a instance 'e' to be a concept 'c'),
- BLC (Basic-Level Categorization),
- MI (mutual information),
- NPMI (Normalized Pointwise Mutual Information) and,
- PMI∧K (corresponds to NPMI).

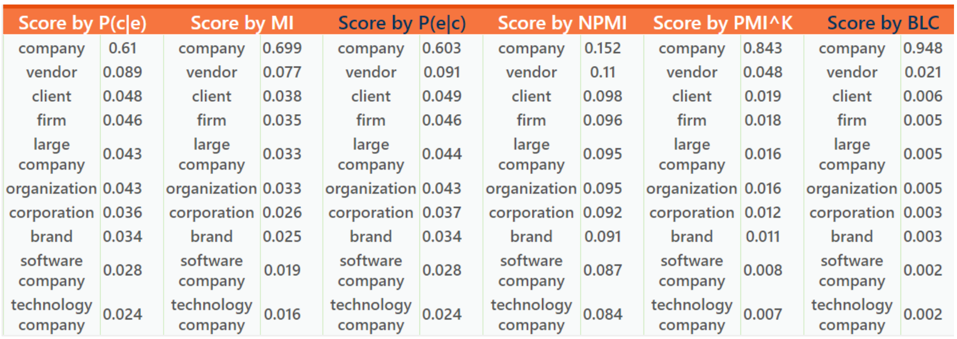
BLC, MI, NPMI, and PMI∧K are all derived from:

- P(c|e),
- P(e|c),
- P(e)(probability of instance *'e'*), and
- P(c)(probability of concept *'c'*)

using different calculation methods [56] [58] [59] [60] [61] [62]. The recommended methods are BLC and smoothed P(e|c) [32]. For instance, the instance '*Microsoft*' using BLC gives the concept '*company*' with probability: '*0.948*' . Apart from this, the user can also select the top k candidates for the results. Fig. 2.1 shows the results obtained for the instance '*Microsoft*' with the top 10 candidates and 0.9 smooth.

DBpedia Spotlight

DBpedia Spotlight [63] is a tool of DBpedia which helps in named entity recognition. The tool comprises of three databases namely, DBpedia, Freebase and Schema.org. The user can select types like "*Astronomy*", "*Color*", "*Internet*", "*Bussiness*", etc. from these databases to have only entity recognized of specific fields. Apart from this user can give custom (SPARQL) query as well. User can set a confidence degree as well. Confidence states the truthfulness of label for entities. It can be set between the range of [0-1]. Confidence is usually inversely proportional to the number of identified entities, i.e with less confidence more entities can be recognized but with lower accuracy.



Score by P(c|e)		Score by MI		Score by P(e|c)		Score by NPMI		Score by PMI^K		Score by BLC	
company	0.61	company	0.699	company	0.603	company	0.152	company	0.843	company	0.948
vendor	0.089	vendor	0.077	vendor	0.091	vendor	0.11	vendor	0.048	vendor	0.021
client	0.048	client	0.038	client	0.049	client	0.098	client	0.019	client	0.006
firm	0.046	firm	0.035	firm	0.046	firm	0.096	firm	0.018	firm	0.005
large company	0.043	large company	0.033	large company	0.044	large company	0.095	large company	0.016	large company	0.005
organization	0.043	organization	0.033	organization	0.043	organization	0.095	organization	0.016	organization	0.005
corporation	0.036	corporation	0.026	corporation	0.037	corporation	0.092	corporation	0.012	corporation	0.003
brand	0.034	brand	0.025	brand	0.034	brand	0.091	brand	0.011	brand	0.003
software company	0.028	software company	0.019	software company	0.028	software company	0.087	software company	0.008	software company	0.002
technology company	0.024	technology company	0.016	technology company	0.024	technology company	0.084	technology company	0.007	technology company	0.002

Figure 2.1: Single Instance Conceptualization example - Microsoft Concept Graph

Cortical.io

Cortical.io [64] is a software for extracting and analyzing keywords. The software development is inspired by the theory of "*Semantic Folding*" which is, encoding the semantics of natural language text in a semantically grounded binary representation [65]. A semantic space called "*Retina Database*" is built using unsupervised learning method on different categories of documents, including textbooks, customer reviews on various topics, etc. Then, the text is converted into its numerical representation to capture its semantic meaning, this is termed as "*Semantic Fingerprint*". In this process 16,000 features are captured for every word. The relatedness between words is obtained by the overlap in their semantic fingerprints [66]. The model can be trained for any domain specific vocabulary for understanding the text in sentences, paragraphs and documents. The full potential of this software can only be unlocked when it is purchased.

MonkeyLearn

MoneyLearn [67] is a closed domain software for extracting keywords and key phrases from text data. The model is built using deep learning methods which can be trained on domain specific data. It includes entity extractor and classifier. The extracted keywords can be tagged according to its classification. The API of this model can be integrated with five programming languages namely, Python, Java, JavaScript, Ruby and PHP.

Apart from the above mentioned models, some famous platform for NLP are IBM Watson Natural Language Understanding [68], Amazon Comprehend [69] and TextRazor [70].

2.3.3 Related Work

It would not be wrong to say that Named Entity Recognition built a platform for Concept Extraction. It was only after the emergence and advancement of Named Entity Recognition that Concept extraction came into picture.

"*The named entities are linguistic expressions that denote ontological objects*" [34] in data sources. The task of NER was first introduced at sixth MUC [71] in 1995, which involved recognition of names, numerical values and temporal expressions, and then categorizing them under the following labels:

- ENAMEX comprises of names and acronyms of person, organization or geographical locations,
- TIMEX includes date and time expressions, and
- NUMEX includes money, percentage and quantity.

Since then, researchers have been working on the data of various general (like, news articles or blog text) and specific (like, biomedical) domains define NER and Concept Extraction as key tasks for information extraction. [53] aims at extracting the relevant concepts from blog texts to populate a information repository. A domain specific ontology is developed to extract information in form of triplets (subject, predicate and object). The subject of the sentence is treated as a concept which is identified as a noun phrase; the predicate is treated as an attribute which is the verb phrase and the object is the attribute value which is either a proper noun or an adjective phrase. The "*theme concept*" is defined as the subject in one or more sentences and the one that occurs the maximum number of times; the domain of theme concept is specified either by its explicit mention or by matching the attributes stated in the blog text with an attributes' list. The attribute list is made with the help of domain experts and attribute values are extracted using pattern matching. The approach is to extract the theme concept, match it with the domain and extract attribute values to update or populate the domain. Different algorithms are implemented using Java for triple extraction from parse tree. The parse tree is obtained using Stanford CoreNLP [72] Java library.

[54] detects the biomedical concepts as complete noun phrases or adjective phrases. Author states three categories of concepts in the domain of this research namely, medical problems, tests and treatments. Preprocessing of the data includes normalization, POS tagging and then mapping it to the concepts of the Unified Medical Language System's database. [51] extracts clinically related entities from electronic health records using dictionary matching. Three approaches are used to extract them from the free text. First, word to word matching over the whole sentence is done. Second,

each word in a sentence is lemmentize and converted to its base form and then matched with the dictionary. Third, the spelling of words are compared with the dictionary terms using "*edit distance*" approach. *(Edit distance is a string similarity measure, it refers to the number of changes required to convert one string to another.)*

[52] extracts information related to drug crimes, it includes drugs' names, price and quantity, drug dealer's nationality and drug hiding methods. The researchers constructed a list of drugs' names, price, quantity, nationalities and drug hiding methods and extracted the same by matching the texts with the specific lists. Preprocessing of the text includes tokenization and POS tagging. The Stanford CoreNLP [72] is used to preprocess the text. Then different grammar rules and matching algorithms were constructed for extracting and identifying different entities. The drugs' names, drug dealers' nationality and drug hiding methods are identified either when the extracted term matches exactly with any of the term in the list or when the extracted term combined with its previous term match with the term in the list. Prices and quantities are identifies if the extracted token is a number and the token before or after the numerical value matches with the list of the same.

Most of the works described in this chapter lack annotated data as either it is not present or it is costly to obtain. Systems Engineers Virtual Assistant (SEVA) for NASA [49] is built without the use of labeled data. SEVA is used by system engineers (SE) to assist them in keeping records of information to solve problems of NASA specific projects and answer user's queries. [49] states that, in comparison with general natural language free text, SE concepts are less ambiguous. One concept usually refers one thing. For instance, the word *code* means the same when referring to a *programming code*, *code modification*, or *code integration*' in the SE domain. A "*System Engineers'*" handbook along with domain experts were used to make the dataset of concepts. Here, also, the concepts are majorly defined as noun phrases. Python library NLTK and RegEx are used to make pattern matching in order to detect SE's concepts in user queries. The entities are constructed with the aid of POS tags by iterating in the increasing order of number of words in the definition. This process helps in creating the hyponymy [73] between "*a lower-word entity and a higher-word entity*"; for example, '*Technical Requirements*' is the subset of '*Requirement*'. In this process every root entity is lemmatized. Further, the concepts are classified into categories and the model is tuned using BERT language model and Pytorch.

With the availability of labeled data many deep learning methods can be used to identify entities [74] [75] [76].

2.4 Relation Extraction

Relation in the context of NLP is defined as a connection that exists between entities. For example, in the sentence '*Charles Babbage was born in London.*' the association between the entities '*Charles Babbage*' and '*London*' is '*born in*'. Relation extraction is a sub-task of information extraction, that aims at identifying relations or relational phrases from text data. "*In essence, it allows us to acquire structured knowledge from unstructured text*" [77]. Usually the focus of the relation extraction task is to detect predefined relations among entities, like extracting '*works at*' relation between employee and organization or '*lives in*' relation between person and location. These types of relations are known as binary relations. Relational phrases can be seen as the surface from which relations are extracted [78].

2.4.1 Tools and approaches

Relations can be extracted from using general rule based methods [49] [79] [80] [81] to complex deep learning methods. The majority of the recent work for relation extraction is supervised [82] [83] [84] [85] and semi-supervised [84] [85] methods. Unsupervised methods like Open IE [86] which do not require any annotated data but rely on defined constraints. As unsupervised methods dependent on pre defined rules/constraints and, in some cases even small set of labeled data to push start the system, it is often questioned if this is actually unsupervised [87]. Rule based approaches apply hand crafted rules for extracting specific patterns of relations from the text. POS tags along with regular expressions are used for building the patterns. Apart from this, relations are also generated manually, like making a list or dictionary of relation and then finding the entities that are connected via it. For instance, it is very common to create a relation '*stands for*' to connect abbreviations with its full form.

Python libraries Spacy [23] and NLTK [24], and Stanford's CoreNLP [72] Java library are some famous tools for POS tagging and generating grammar rules. Some well known models for relation extraction includes, MonkeyLearn [67], Amazon Comprehend [69], IBM Watson Natural Language Understanding [68] and Cortical.io [66] [64]. MonkeyLearn and Cortical.io are specific for extracting keyword and key phrases, and they can symbolize relations as well as concepts.

2.4.2 Related Work

The task of relation extraction is very famous, and every day new approaches are emerging and new researches are being made in this field. A major application domain of relation extraction in biomedical data is to extract relationships between entities like gene-disease [50], problem-treatment, test-treatment [54], and many more. [50] detects diseases and their corresponding genes from journal articles by constructing a gene and a disease dictionary. When a gene is extracted with a corresponding disease then it is termed as a relation. Both, the gene and the disease are extracted using the rule of longest dictionary match. The dictionaries are constructed by combing the information scattered in different public databases of biomedical sciences. Then, an annotated data set is created. The annotated data is fed to a maximum entropy model to improve the results. [88] extracts relation as events that occur between bio-medical entities, from literature data. Rule based approach is used to construct the model. A Dependency Parse tree is used for POS tagging. The authors have constructed three rules for extracting the relations. First rule extracts a relational term that occurs between bio-medical entities, both the relation and entity terms are tagged as noun phrases, it is called the "*binding event*". Second rule is built to extract the relation term which signifies movement, it occurs after en entity and before the general structure of phrase '*from location to location*'. Third rule extracts the relation term which occur as verb phrase between the first and second rule.

[65] extracts information related to biomedical domain from unstructured free text. For extracting the relations which are specific for biomedical domain researchers have first constructed a dictionary, and then constructed rules for extracting the relations. Relational phrases are defined as verb phrases which occur between entities. Dependency parse tree is used for POS tagging and, preprocessing of the text includes normalization. Then the relations are classified into four categories. [53] [74] [75] [76] [54] take advantage of labeled training data to identify relations using various neural networks.

In comparison to biomedical domain, significantly less research has been done to identify relations from any other domain specific (like telecommunication) data. SEVA [49] uses hyponyms [73] and verb phrase chunking for extracting relations from user queries that are specific to NASA's projects. Homonym's are created to identify more specific entities using the noun tag that surround already identified concepts. For example, in a sentence *SE functions should be performed*, *'SE'* has a tag NNP (proper noun, singular) and *'functions'* has a tag NNS (plural noun) thus hyponymy *'subset of'* is created between 'SE functions' and 'SE'. A relation "*stands-for*" is created to link abbreviations with its full forms. The whole procedure helps in constructing a knowledge graph. For extracting relations by verb phrase chunking, triples from sentences are constructed, using entities extracted from the CRF model and definitions. For extracting only the verb phrases that join two entities, grammar rules are build using NLTKs regex parser and chunker.

With this, relation like phrases are extracted from the sentences. For example, in the sentence: '*System requirements are understood by the programmers*' , the entities '*System requirements*' and '*programmers*' are connected by the relation '*understood*'.

[89] [90] have developed are some high performing neural network models for extracting relations from the New York Times Corpus [91].

Chapter 3

Proposed Methodology

The major aim of this thesis is to develop a system that can extract domain specific information from natural language free text. This chapter describes the system architecture of the proposed system for domain specific information extraction. Along with this, the chapter also details the tools and approaches that were used to develop the same.

3.1 System Architecture

We propose a system for extracting domain specific information from unstructured text by identifying and extracting related concepts and relations, and then, combining their output to present as a final result. The system unit consists of four modules namely:

- *Input Processing unit*,
- *Concept Extraction module*, which extracts the specific concepts from input text,
- *Relation Extraction module*, which extracts the relational phrases from input text, and an
- *Output Processing unit*, which process the output of concept and relation extraction module to form a result.

Figure 3.1 shows the block diagram of the system architecture. The example sentence in the Figure, and example sentences used later in this chapter are taken from a report of NASA's mission *Stardust* [92].

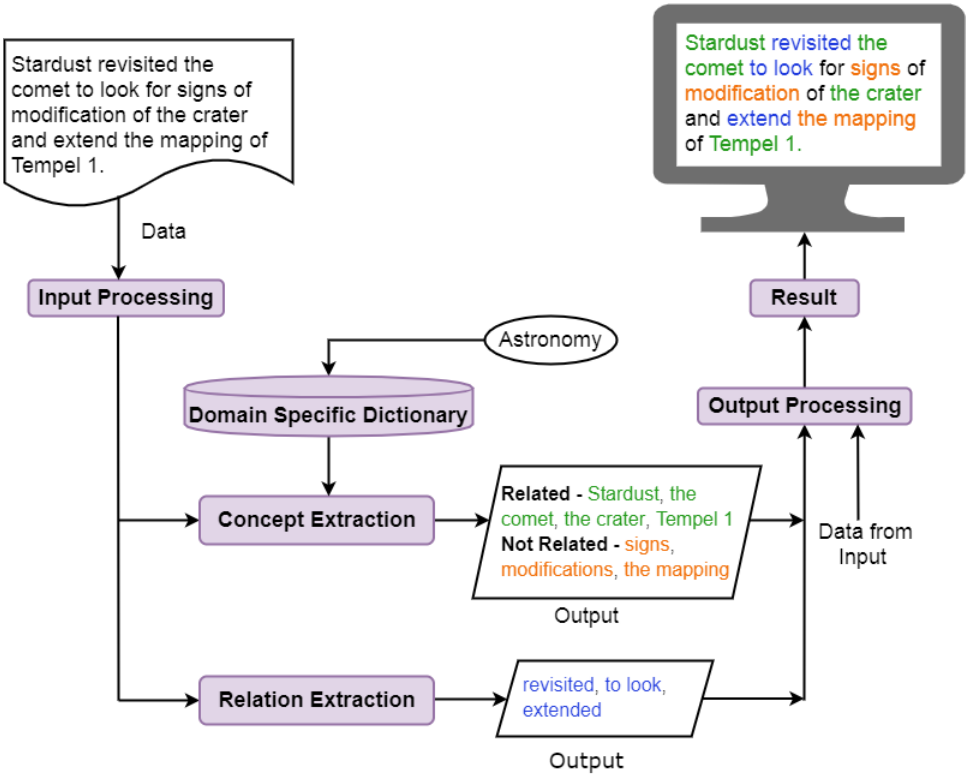

Stardust revisited the comet to look for signs of modification of the crater and extend the mapping of Tempel 1.
Data
Input Processing
Astronomy
Domain Specific Dictionary
Concept Extraction
Related - Stardust, the comet, the crater, Tempel 1
Not Related - signs, modifications, the mapping
Output
Relation Extraction
revisited, to look, extended
Output
Data from Input
Output Processing
Result
Stardust revisited the comet to look for signs of modification of the crater and extend the mapping of Tempel 1.

Figure 3.1: Block Diagram of System Architecture

3.1.1 Input Processing

The input processing unit takes text as an input, detects the language of the text and based on the result the text is passed on to the next unit .The input can be sentences or paragraphs composed in a file. The user can give any amount of inputs in the input file.

Language Detection.

A natural language text can be presented in any language. Language detection is important, as all the languages follow different grammar rules and have different language models. This step also gives user the freedom, as well as, scope to have any and all languages as an input text.

The language detection function will detect the language in which the input is provided and based on its result the text will be passed on to the next function.

3.1.2 Concept Extraction

The concept extraction module extracts domain specific concepts from input texts. It is dependent on the dictionary for identifying the concepts that are needed to be extracted. The dictionary plays the part of a knowledge base as it consists of keywords or key phrases that are to be identified. This module is flexible to the dictionary, meaning, if provided with any specific / general purpose dictionary, the system will extract related phrases / words from the text according to that dictionary. The module can also checks for entries of extracted phrases on DBpedia database.

The concept extraction module comprises of three functions namely: Noun Phrase Extraction, Phrase matching with dictionary and Phrase matching with DBpedia. Fig.3.2 shows the block diagram of the architecture of the proposed concept extraction module.

Noun Phrase Extraction.

It was studied, and analyzed that most of the concepts or entities mentioned in text are of the type Noun Phrases [49] [53] [74]. Therefore, in order to identify concepts, noun phrases need to be extracted. This function extracts noun phrases from every input entry separately. After noun phrase extraction, they are matched with the dictionary, and if DBpedia is enabled by the user, then the extracted phrases are matched with DBpedia Database as well.

Noun Phrase matching with dictionary

The next in pipeline is dictionary matching. The extracted noun phrases are matched with the dictionary two parts namely: Complete Match and Partial Match. They are as follows:

- Complete Matching with Dictionary
  The complete match function checks if the extracted noun phrases match exactly with phrases from the dictionary. If a complete match is found then they are saved as 'Related' and will be added to the output file.

- Partial Matching with Dictionary
  If the noun phrase is not matched completely with the dictionary then the system tries to find a partial match. The partial match of the phrase is done in three steps: first, tokenize the phrases into words; second, convert words to their base form by lemmatization; third, match

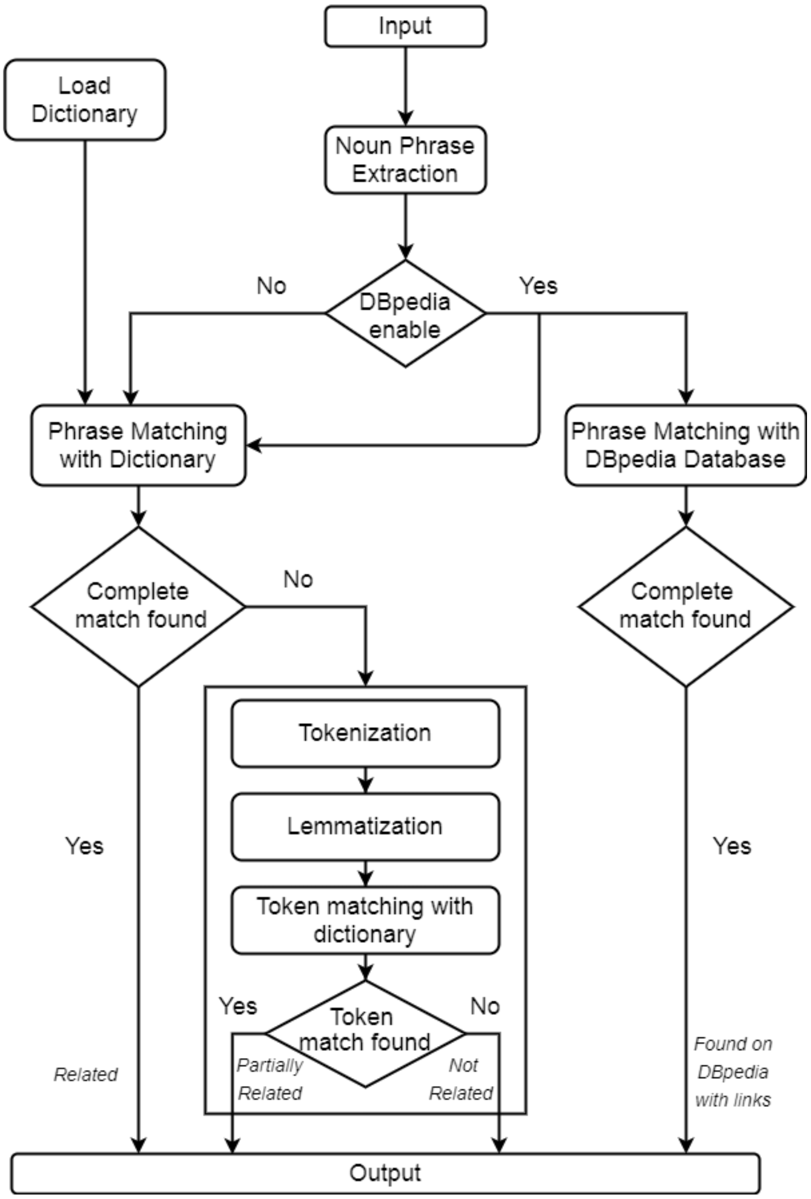


Figure 3.2: Block Diagram of Architecture for Concept Extraction

each word lemma from the phrase with the dictionary. If at least one word lemma of the phrase matches a word in the dictionary. Then, the whole phrase is saved in an array of partially related terms to latter be seen in the final results. For example, given an Astronomy dictionary, the phrase '*the comet's mass*' is recognized as a partial match as the term '*comet*' is in the dictionary.

The phrases who dont fall in any category are also saved in a not related term array.

Noun Phrase matching from DBpedia.

The extracted noun phrases can be matched with DBpedia [93] knowledge base to check if they have a link on DBpedia database. DBpedia knowledge base consists of all kinds of entities including persons, creative work, organizations, and etc. As DBpedia provides linked data, SPARQL queries are used for pulling out information from it. If the complete phrase matches with the DBpedia database entries then they are saved with its corresponding link and added to the output file.

3.1.3 Relation Extraction

For detecting the relations from the sentences, we propose the extraction of verb phrases from the input text.

Verb Phrase Extraction.

It was studied that verb phrases contain relational information that help in linking entities [49][50]. This module tries to extract all the relational phrases from the sentences and not only the ones that connect entities. This because, it was observed that some of the information that help in understanding the text gets ignored by only extracting relations between concepts.A second reason is, as the text is unstructured in nature they may not always follow a specific pattern and are ambiguous. For example in the sentence '*Comets were formed at the same time as the solar system and are made up of primitive condensates and grains incorporated into them at this time.*', given an Astronomical Sciences dictionary we get:

$$Concepts : \{\textit{Comets, the solar system}\}$$

If only relational phrases between them are extracted, then only one result will be obtained, that is:

$$relations : \{\textit{wear formed}\}$$

However by extracting all relational phrases we get:

relations : {*were formed, are made, incorporated into*}

3.1.4 Output Processing

The output obtained from Concept and Relation Extraction modules are independent of each other. The Output Processing module takes the output of the both the modules and the original text as its input. It then matches the outputs with the original text string and tag them accordingly. The tagged text is then displayed as the final result.

3.2 Tools and Approaches for developing the Proposed System

This section describes the tools and approaches that were used for developing the proposed system. The system extracts information from English Language free text. It is built using Python [94] [95].

3.2.1 Input Processing

Language Detection.

For detecting the language of the input text, Python open source library named *"langdetect"*[96] is used. Langdetect library is a very vast library, as it supports 55 most spoken languages, including Portuguese and Spanish. If the detected language is English the texts is processed further else no processing is done and a message is passed on to be displayed in the Result saying "*The system works only for English.*"

3.2.2 Concept Extraction

Dictionary Loading.

The dictionary is loaded as an Excel spreadsheet. To read the excel sheet python's library *"pandas"* [97] is used. Pandas is a tool that is used for handling and analyzing all kinds of data from tabular to multidimensional.

Noun Phrase Extraction.

The extraction of noun phrases is accomplished by using *"Spacy"* [98] a python library. 'Spacy' is a very useful and handy library for natural language processing. It is one of the fastest tools in the world for NLP [98]. An English model *"en_core_web_sm"* [35] is used for tagging noun phrases. The language model ' en_core_web_sm' is an English multi-task CNN trained on OntoNotes.

Phrase Matching with Dictionary.

For finding the phrases that have a complete match with the dictionary, string matching of noun phrases with dictionary terms are done in linear fashion. If the string is matched then it is saved in an array. For finding the partial match, tokenization and lemmatization are done with the help of 'Spacy' and 'en_core_web_sm'. Each token of the phrase is matched with dictionary terms via string matching.

Phrase matching from DBpedia.

4.58 million things are described in English version of DBpedia Database [93]. To obtain the DBpedia links for the extracted noun phrases, a Python library *"SPARQLWrapper"* [99] is used. 'SPARQLWrapper' is a tool which helps in creating query strings on DBpedia and alter the results obtained in easily understandable formats.

3.2.3 Relation Extraction

Verb phrase Detection.

To understand the structure of patterns, two methods were used. First, analyzing the general structure of English sentences and second, understanding the structure of queries. To understand the structure of general English sentences, assistance is taken from the websites [100][101] and based on that 8 patterns were constructed. These 8 patterns suffice in extracting verb phrases from majority of the examples listed in the websites. However, as the free text does not always follow the grammar rules, 30 sentences were picked in random fashion from the dataset, and their structure was analyzed. After analyzing those sentences, 7 more patterns were added for extracting verb phrases. The total of all 15 patterns that were used for extracting verb phrases are listed in Appendix B.

Verb phrase Extraction.

A rule base approach is used extracting verb phrases. Python's Library 'Spacy' is used for creating and matching the rules. The builtin function *'Matcher'* [102] in 'Spacy' allows to create patterns that are needed to be extracted from the sentences and, a built-in function *'Matcher.add'* [103] allows to add multiple matching patterns.

Overlapping Score.

While evaluating the results for relational phrase extraction it was observed that some of the relational phrases overlap with concept phrases. For example, the complete phrase '*newly installed FTTH* is detected as noun phrase and '*newly installed*' is detected as relational phrase. This happens because noun phrases contain Adverb and Verb tags alongside with Noun tag. The patterns made for verb phrase extraction does not contain any Noun tag. To understand this issue a formula is deduced to calculate the overlapping score of verb phrases. It is as follows:

$$\text{Overlapping Score} = \frac{\text{Number of Overlapping words in verb phrase}}{\text{Total number of words in verb phrase}}.$$

3.2.4 Output Processing

Matching and Tagging Output.

The output from both Concept and Relation Extraction module are matched with the original text. The entities that were in Related and Partially Related array are tagged as *'Related'* , the entities that were in Not Related array are tagged as *'Not Related'*, and the relational phrases are tagged as *'Relation'*. For matching and tagging the string python's module *"re"* [104] is used. The module 're' matches regular expressions in a string. The 're' function '.sub' helps in modifying the string by replacing the occurrence of preferred substring with a desired substring.

3.3 Tools and Techniques for comparative analysis

This section describes the methods that were used for the comparative analysis of proposed concept extraction module with available Concept Extraction Platforms. Two platforms were used for this analysis, namely: Microsoft

Concept Graph and DBpediaSpotlight.

3.3.1 Microsoft Concept Graph

The Microsoft Concept Graph provides single instance conceptualization therefore, from the sentences, noun phrases are extracted and checked for complete and partial dictionary match. The complete and partial dictionary matched phrases are saved in different arrays. Both partial and complete match phrases are checked for entries on Microsoft concept graph. For this, the recommended BLC and smoothed $P(e|c)$ were used to extract the top 3 candidates for the found phrases. Top 3 candidates were selected as, some phrases may give wrong results in the first concept due to its terminology; for instance '*mouse*' in queries is used as a computer device but in the graph the first and most probable is '*animal*' and the second is '*computer device*'.

The Microsoft concept graph provides results in form of XML files. For pulling the data, Python's library *"urllib"* [30] is used. This library helps in opening the URLs and retrieving the data in a dictionary data type. This experiment is done in two parts, first without preprocessing the extracted phrases and second with pre-processed extracted phrases. The matched phrases of all the queries (both complete and partial) were kept in two different arrays. All the duplicates were discarded and only unique strings were taken into consideration. Then with the 'urlllib', each phrase is checked in linear fashion. In first part, the phrases are checked as they were stored in the array; in the second part of the experiment, the extracted phrases are preprocessed in 3 steps. First, all the stop words from the starting of the string are removed: for example- '*the public ip*' is converted to '*public ip*'. Secondly,all the words in the phrase are lemmatized to its base form. Lastly, the array is again checked for duplicate entries and all the duplicates are discarded to keep only unique phrases. The duplicates were checked again as, there were phrases like '*notifications*' and '*the notification* which were treated as two different strings without preprocessing, however after the stop word removal, and the conversion of word to its base form, they both were changed to '*notification*'. As, both the complete and partial match phrases are set in different arrays so their counts of how many were associated with Microsoft Concept Graph are done separately. In the end, the results are calculated by the equation given below, for complete and partial match phrases and both of them together.

$$result = \frac{\text{Terms found on Microsoft Concept Graph}}{\text{Total Terms}}$$

3.3.2 DBpedia Spotlight

DBpedia spotlight is a platform of DBpedia which helps in named entity recognition [63]. This platform accepts the whole sentence as its input. The tool comprises of three databases namely, DBpedia, Freebase and Schema.org. The user can select types like "*Astronomy*", "*Color*", "*Internet*", "*Bussiness*", etc, from these databases to have only entity recognized of specific fields. Apart from this, user can have custom SPARQL query as well. The demo is this platform is used for the comparative analysis. To make a comparison of accuracy for recognizing concepts between DBpedia Spotlight and the proposed concept extraction module, example English sentences from the dataset are taken, and then, the recognition is made via both the tools.

Chapter 4

Results and Evaluation

This chapter discusses the results obtained from the developed system using Altice Lab's data. Along with this, a comparative analysis between concept extraction platforms and the concept extraction module is discussed as well.

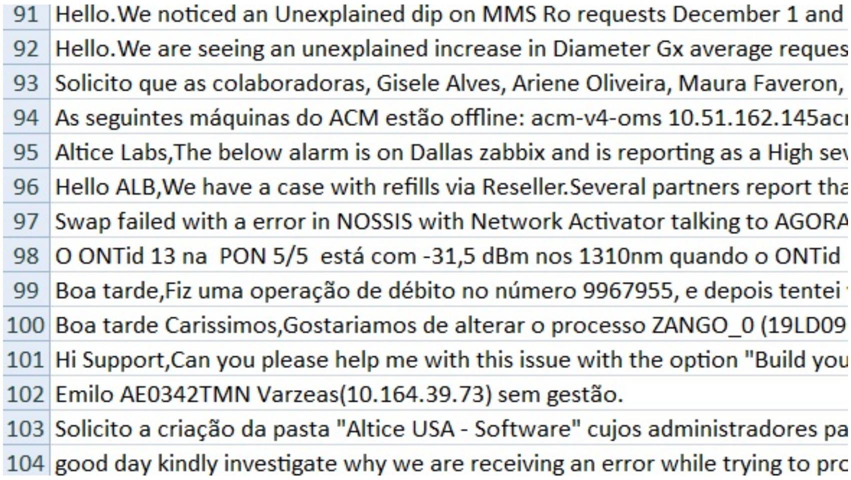
4.1 Case Study on Altice Lab's Data

Altice Lab's data was used for experimenting and evaluating the proposed system. This case study is divided into four sections. The first section describes the Altice LAb's dataset. The second section describes the dictionary that was created to accomplish the need of having a domain specific dictionary for the targeted dataset. The third section shows the results obtained from concept and relation extraction, and the final result of the system. The fourth section shows the evaluation of the results obtained.

4.1.1 Dataset Overview

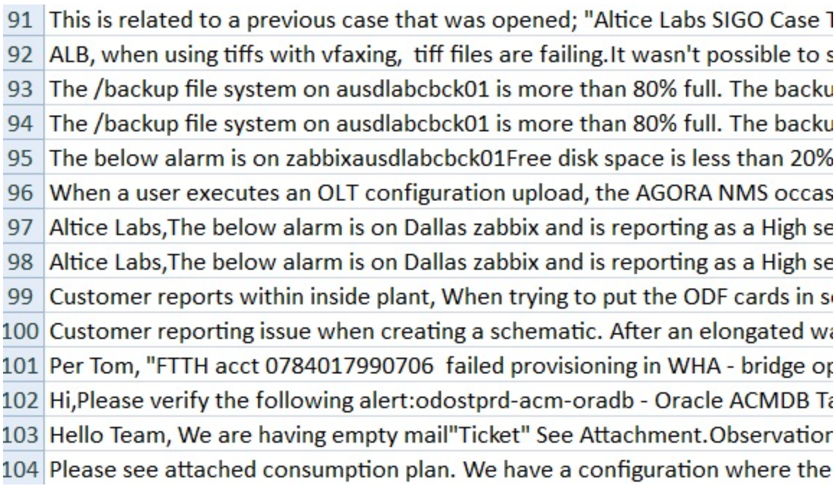
The Altice labs's data was made available to us in November, 2019 in form of Excel spreadsheet. The data consists of real time queries regarding their system. It is a semi-structured data type. It contains a total of 34 columns. The data of one column named *"DESCRICAO"* was extracted and used for the work. It contains the description of the customer report / query. The data in this column is textual. The text is multilingual in nature, comprising of free text in English, Portuguese and Spanish. There were a total of 18,543 rows and the "DESCRICAO" column contains text with one or more sentences. For conducting the experiments, the rows that contain description in English were separated and saved in an another excel sheet.

A total of 7782 rows of free text in English language were bagged after separating text by language. Fig. 4.1 is a sample screenshot of the data with multilingual text. Fig. 4.2 is a sample screenshot of data after separating English text. For the ease of understanding, both the screenshots are taken at the same sequence number of rows, from 91 to 104.



91	Hello.We noticed an Unexplained dip on MMS Ro requests December 1 and
92	Hello.We are seeing an unexplained increase in Diameter Gx average reques
93	Solicito que as colaboradoras, Gisele Alves, Ariene Oliveira, Maura Faveron,
94	As seguintes máquinas do ACM estão offline: acm-v4-oms 10.51.162.145ac
95	Altice Labs,The below alarm is on Dallas zabbix and is reporting as a High se
96	Hello ALB,We have a case with refills via Reseller.Several partners report th
97	Swap failed with a error in NOSSIS with Network Activator talking to AGOR
98	O ONTid 13 na PON 5/5 está com -31,5 dBm nos 1310nm quando o ONTid
99	Boa tarde,Fiz uma operação de débito no número 9967955, e depois tentei
100	Boa tarde Carissimos,Gostariamos de alterar o processo ZANGO_0 (19LD09
101	Hi Support,Can you please help me with this issue with the option "Build you
102	Emilo AE0342TMN Varzeas(10.164.39.73) sem gestão.
103	Solicito a criação da pasta "Altice USA - Software" cujos administradores pa
104	good day kindly investigate why we are receiving an error while trying to pr

Figure 4.1: Data with Multilingual Text



91	This is related to a previous case that was opened; "Altice Labs SIGO Case
92	ALB, when using tiffs with vfaxing, tiff files are failing.It wasn't possible to
93	The /backup file system on ausdlabcbck01 is more than 80% full. The backu
94	The /backup file system on ausdlabcbck01 is more than 80% full. The backu
95	The below alarm is on zabbixausdlabcbck01Free disk space is less than 20%
96	When a user executes an OLT configuration upload, the AGORA NMS occas
97	Altice Labs,The below alarm is on Dallas zabbix and is reporting as a High se
98	Altice Labs,The below alarm is on Dallas zabbix and is reporting as a High se
99	Customer reports within inside plant, When trying to put the ODF cards in s
100	Customer reporting issue when creating a schematic. After an elongated w
101	Per Tom, "FTTH acct 0784017990706 failed provisioning in WHA - bridge o
102	Hi,Please verify the following alert:odostprd-acm-oradb - Oracle ACMDB T
103	Hello Team, We are having empty mail"Ticket" See Attachment.Observatio
104	Please see attached consumption plan. We have a configuration where the

Figure 4.2: Data with English Text

4.1.2 Dictionary Creation

Information technology (IT) related concepts are to be extracted from Altice Labs' dataset. To accomplish this task, a dictionary was created that consists of 994 unique information technology phrases.

The basic need for creating a dictionary is to have a knowledge base that is specific to information technology. Thus, for extracting the IT related words / phrases from the free text of "DESCRICAO" section, this dictionary is created. At present there are many websites available that contain IT glossary, however none of them is extensive at stand alone. The created dictionary is more wide, and since is a .xlsx file(offline) it reduces the processing time of the system.

In order to create this dictionary two methods have been used, namely: Web Scrapping and Manual checking and adding terms. The websites used for crawling IT related terms were [105], [106], [107], [108], [109], [110]. A total of 931 IT terms and phrases were comprised from these websites. After this, a test run is made on randomly selected 150 rows of English text to identify IT related words. Based on the results of the test run, 38 more terms were added to the dictionary. Finally, 25 more terms related to earlier 38 from the test run were added. For example, 4G and GB were the terms among 38 terms so, 2G, TB and MB were added to the dictionary. Thus, by applying these simple yet effective techniques, an extensive dictionary of IT vocabulary was formed.

4.1.3 Results

From 7782 English reports, the system was able to extract 125,182 entities which include:

- Completely Matched IT related terms - 5,319,
- Partially Matched IT related terms - 22,980,
- Not IT related terms - 96,883 and,

a total of 56,351 relational phrases.

Results for Concept extraction

The results obtained for concept extraction are of two types, namely:

- with dictionary matching, and
- with dictionary and DBpedia matching.

It totally depends on user how to save the output from above two options. Each input corresponds to both the types of output. The following are the examples of the results.

Input- Table 4.1 lists example queries in English language taken from the data set.

Query No.	Query Description
1	Customer MJOHNS17 - reports - When placing a PDO and choose the supplier location, if user adds it by pressing the enter key it will not place the PDO. If user clicks the headend with the mouse it goes right through. Issue occurs with all browsers, including FireFox. Asked Mark to try in UAT as well. Said ...Just tried it in the test, same issue
2	Hi Alticelabs,Please help us on checking why the reporting platform isnt working. It says no data for any report que try to generate. Thanks and best regards, LD

Table 4.1: INPUT

First, results of concepts extracted only from dictionary matching are shown and then for dictionary and DBpedia matching are shown.

Output with dictionary matching. The results obtained from concept extraction module with dictionary matching for both the queries in Table 4.1 are shown in table 4.2.

Output with dictionary and DBpedia matching. The results obtained from dictionary and DBpedia matching for both the queries from Table 4.1 are shown in table 4.3.

Results for Relation Extraction

The results for relation extraction methods are of two types, namely:

- No overlapping between concept phrases and relational phrases and,
- overlapping between concept phrases and relational phrases.

Query No.	Related	Partially Related	Not Related
1	user, FireFox.	the enter key, the mouse, all browsers.	Customer MJOHNS17 - reports, a PDO, the supplier location, it, the PDO, the headend, Issue, Asked Mark, UAT, the test.
2		the reporting platform, no data	us, It, any report, que, Thanks, best regards

Table 4.2: Output with dictionary matching

Query No.	Related	Partially Related	Not Related	Found on DBpedia
1	user, FireFox.	the enter key, the mouse, all browsers.	Customer MJOHNS17 - reports, a PDO, the supplier location, it, the PDO, the headend, Issue, Asked Mark, UAT, the test.	User (*User), It (*It), Issue(*Issue), FireFox (*FireFox)
2		the reporting platform, no data	us, It, any report, que, Thanks, best regards	us (*us), It (*It), que (*que), Thanks (*Thanks), best regards (*best regards)

*here, * = http://dbpedia.org/resource/*

Table 4.3: Output With dictionary and DBpedia matching

Table 4.4 shows the results where there is no overlap between relational and concept phrases

Query Description	Relations
The SMS notifications for landing page redirections are not being generated. They should be generated anytime the customer is browsing through no HTTP traffic and doesn't have a valid package.	landing, are not being generated, should be generated, is browsing

Table 4.4: Relation phrases with no overlap

Table 4.5 shows sample queries that contain relational phrases which overlap with concept phrases.

Query No.	Query Description
1	Hi Support, Can you please check sitecode attribute filter? It seems it isn't working. Attach there is a screenshot showing alarms for site 204, but in the top window shows not alarm. Thanks for your help. Regards, Hansell F.
2	We had a tech installing a ONU that was not broadcasting wifi networks.. Tech swapped ONU and issue persisted. Tech advised WPS light is not on. Tech can get online when hard wired to ONU, but unable to see/join any wifi networks. Please see attached images for wifi status. When viewing account in NetQ wifi status appears as Dormant.
3	The user has problem with creaton of new backup file. Incident is urgent because it block all planified operation. We made stop / start of application, but without succes. The button Create new backup file is still missing. Could you check, please?
4	The customer unable to reach the internet. Provisioning is showing no issues / errors.
5	Hi Support: We are involved in a new MSC and ATS destination and we need your assistance to define this configuration in PCC, See attached the information needed, this topic is high priority, let me know as soon as you have some update.BR OC

Table 4.5: Relational phrases overlapping with concept phrases - sample Queries

All the above mentioned 5 queries contain different degree of overlap. Table 4.6 shows the different results that were obtained when relational and concept phrases overlap along with its overlapping score.

Query No.	**Relational Phrases**	**corresponding concept phrase**	**overlapping word(s)**	**overlapping score**
1	Can, check, seems, isn't working, Attach, *screenshot showing*, shows	screenshot showing alarm	screenshot showing	2/2 = 1
2	installing, was not broadcasting, swapped, persisted, advised, not on, can, online when, unable to see, join, *see attached*, when viewing, appears	attached images	attached	1/2 = 0.5
3	is urgent, *block all planified*, made stop, Create, still missing, check	all planified operations	all planified	2/3 = 0.6667
4	unable to reach, *Provisioning is showing*	Provisioning	Provisioning	1/3 = 0.3333
5	are involved, need, to define, *See attached*, needed, *is high*, know	attached the information, high priority	attached, high	(1/2) + (1/2) = 1

Table 4.6: Relational phrases overlapping with concept phrases

It can be seen from the above examples that there are different degrees at which relational phrases overlap with concept phrases and that there can be more than one phrases in a query that overlap.

When combining the results of concept and relational phrases extraction, the following results were obtained.

Example 1:

network element polling executed by system pt.ptinovacao.agorang.ap.systemmonitor. SystemMonitorBean occasionally does not update internal cache or close open alarms, generating incorrect operational status in OLTs.

Example 2:

At the moment newly installed FTTH customers are unable to browse the internet. Looks like they are missing the following two settings in Agora; OVLAN and IVLAN.We would like to know where Agora gets these two settings from and if the Netwin upgrade (CHG0056354 - Netwin USA 2.3.0) could have caused this issue.If possible please join bridge for assistance as we are currently troubleshooting live customers. Bridge: 516-803-7700 Conf code: 337277. Thanks

The colours used define the following:

- Related IT Concepts(Completely/Partially related)
- Relational Phrases
- Not related entities
- Overlapping word(s)
- text that does not fit under any category.

If the input is given in any other language then a remark is returned as a result along with the text's language ISO 639-1 code [96]. Table 4.7 shows query description in Portuguese language.

Query No.	Query Description
1	Segue erro que impede meu acesso ao site do SOC para agendamento de exames médicos.

Table 4.7: Input in Portuguese language

Output 2- Table 4.8 shows output for Input in Portuguese language 4.7.

Query No.	Language	Note
1	pt	Only Works for English.

Table 4.8: Output for Portuguese language

An excerpt of the result is given in Appendix A.

4.1.4 Evaluation

The evaluation of the results are done manually on 80 sample, randomly selected query descriptions from the English dataset.

Evaluation for Concept Extraction

The evaluation for IT concept extraction from the IT Dictionary is done in form of accuracy, recall and precision. ALL the Completely and Partially IT phrases are taken as True Positive, all the not IT related phrases are taken as True Negative. The phrases which were not IT related but detected as completely/partially IT related are taken as False Positives, and all the phrases that were actually IT related but detected as not IT related or was undetected are taken as False Negatives. Table 4.9 shows the results for the same.

Accuracy	Precision	Recall
91.3642	98.2188	81.0924

Table 4.9: Evaluation of IT Concept Extraction results from Dictionary

Evaluation of Relational Phrase Extraction

The evaluation of relational phrases is done in form of precision and recall. Accuracy was not included as it uses True Negatives in it's calculations and, with relational phrases the formulation of true negatives is not defined.

As relational phrases contains overlapping words and, the calculation of overlapping words does not fit in the calculation of precision and recall. To overcome this, the overlapping scores were formulated in 4 ways which are:

1. All the overlapping phrases are considered as True Positives.
2. All the overlapping phrases are considered as False Positives.

3. If overlapping score is ≤0.3333 then its True Positive else False Positive.
4. If overlapping score is ≤0.5 then True Positive else False Positive.

Apart from that all the relational phrases that were not detected are seemed as False Negatives and all the wrongly detected relational phrases are seemed as False Positives. Table 4.10 shows the results for the same.

+

	Precision	**Recall**
All overlapping phrases as True Positive	98.4344	93.8432
All overlapping phrases as False positive	92.2935	93.9432
If overlapping score ≤0.3333 then True Positive else False Positive	94.1384	93.9560
If overlapping score ≤0.5 then True Positive else False Positive	97.7891	94.1385

Table 4.10: Evaluation of Relational Phrase Extraction

Note : The 30 query descriptions that were used for formulating the rules for verb phrase extraction were not included in the sample 80 query description that were used for evaluation.

4.2 Comparative Analysis with Concept Extraction Platforms

4.2.1 Microsoft Concept Graph

Evaluation of results from Microsoft Concept Graph

The results that were obtained from matching IT dictionary related phrases with Microsoft Concept Graph are discussed in this section. 80 sample English queries were taken from the dataset to conduct the experiment. The smooth values of 0.1, 0.3, 0.5, 0.8 and 0.9 were used to get the results. It was seen that the same results obtained from all the smooth values were same. The results would have differed if top 6 to 10 candidates were taken into consideration. However, for the top 5 candidates the results remained the same. Top 3 candidates were taken into consideration for this analysis.

Table 4.11 shows the results obtained without preprocessing the extracted phrases.

	Total Terms(A)	**Terms found in Microsoft Concept Graph(B)**	**Results (B/A)*100**
Completely IT Related	45	45	100
Partially IT Related	258	48	18.6046
Total	303	93	30.693

Table 4.11: Results From Microsoft Concept Graph without Preprocessing

Table 4.12 shows the results obtained with the preprocessed extracted phrases.

	Total Terms(A)	**Terms found in Microsoft Concept Graph(B)**	**Results (B/A)*100**
Completely IT Related	45	45	100
Partially IT Related	233	103	44.206
Total	278	148	53.2374

Table 4.12: Results From Microsoft Concept Graph with Preprocessing

It can be seen that after preprocessing the text the results are improved by 22.5444%. Due to the lemmatization and stop word removal, the total number of Partially IT related terms are decreased from 258 to 233. From the above results it can be seen that as Microsoft Concept Graph conceptualizes all the terms that posses complete match with the dictionary. However, it does not perform very well with partially related terms which are of fairly greater amount in query sentences.

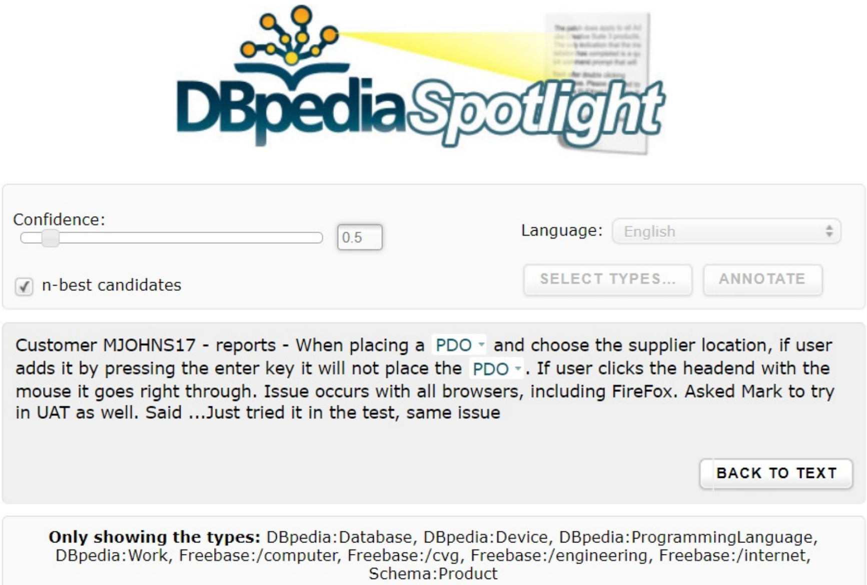
4.2.2 DBpedia Spotlight

As the experiments were specific for IT entity recognition, the following categories were selected from the list of databases:

- DBpedia database: Database, Devices, Programming Language , Work.

- From Freebase database: Computer, Computer and Video Games, Engineering, Internet.
- From Schema.org: Product.

Thirty examples are taken to test with both DBpedia Spotlight and the dictionary we have created. The result of two examples are shown in this section. Fig.4.3 shows the results for example 1 from DBpedia Spotlight at 0.5 confidence, Fig.4.4 shows the results for example 1 from DBpedia Spotlight at 0.1 confidence and Fig.4.5 shows the result from our dictionary.



Confidence: 0.5

Language: English

☑ n-best candidates

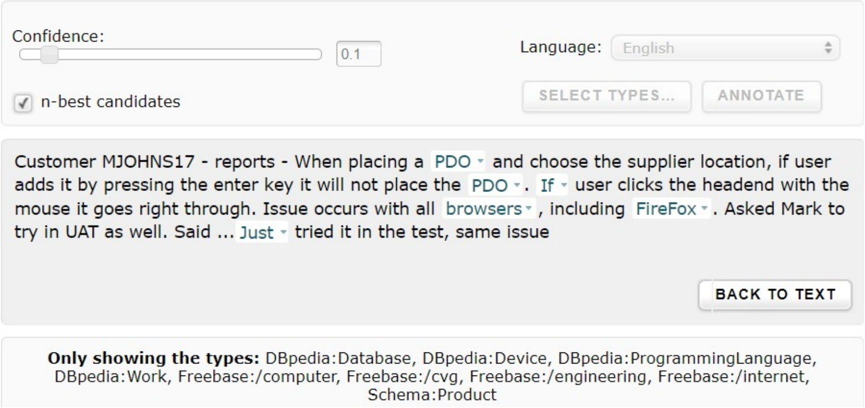
Customer MJOHNS17 - reports - When placing a PDO and choose the supplier location, if user adds it by pressing the enter key it will not place the PDO. If user clicks the headend with the mouse it goes right through. Issue occurs with all browsers, including FireFox. Asked Mark to try in UAT as well. Said ...Just tried it in the test, same issue

Only showing the types: DBpedia:Database, DBpedia:Device, DBpedia:ProgrammingLanguage, DBpedia:Work, Freebase:/computer, Freebase:/cvg, Freebase:/engineering, Freebase:/internet, Schema:Product

Figure 4.3: Example 1 result with 0.5 Confidence from DBPEDIA Spotlight

Fig.4.6 shows the results for example 2 from DBpedia Spotlight at 0.5 confidence. Fig.4.7 shows the results for example 2 from DBpedia Spotlight at 0.1 confidence. Fig.4.8 shows the results for example 2 from DBpedia Spotlight at 0.05 confidence and Fig.4.9 shows the result from our dictionary for example 2.

In Example 1 DBPEDIA Spotlight was able to identify only '*browser*' and '*FireFox*' correctly that to at 0.1 confidence. However concept extraction



Confidence: 0.1

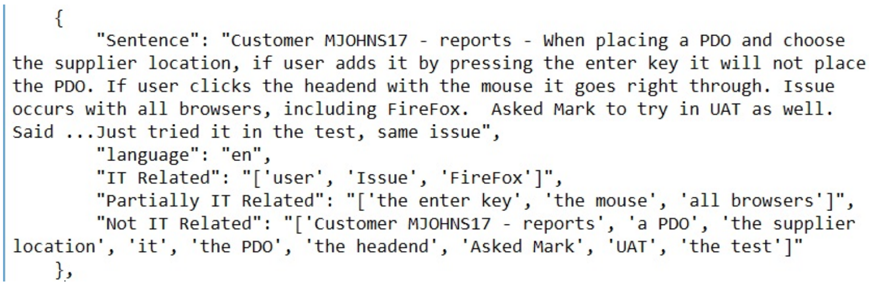
Language: English

☑ n-best candidates

Customer MJOHNS17 - reports - When placing a PDO and choose the supplier location, if user adds it by pressing the enter key it will not place the PDO . If user clicks the headend with the mouse it goes right through. Issue occurs with all browsers , including FireFox . Asked Mark to try in UAT as well. Said ... Just tried it in the test, same issue

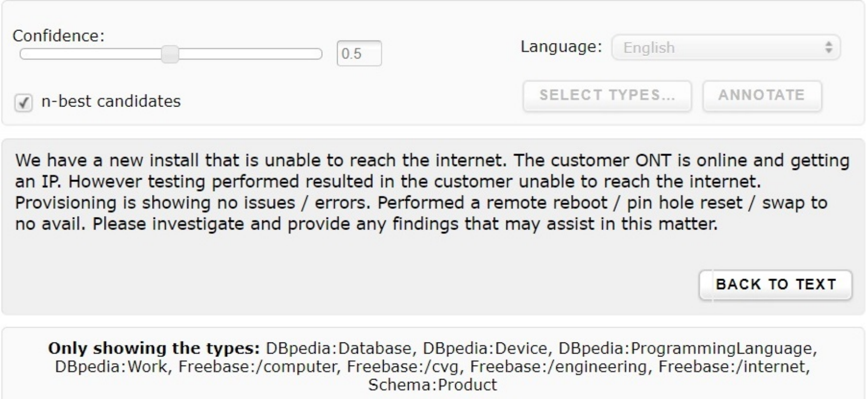
Only showing the types: DBpedia:Database, DBpedia:Device, DBpedia:ProgrammingLanguage, DBpedia:Work, Freebase:/computer, Freebase:/cvg, Freebase:/engineering, Freebase:/internet, Schema:Product

Figure 4.4: Example 1 result with 0.1 Confidence from DBPEDIA Spotlight



```
    {
        "Sentence": "Customer MJOHNS17 - reports - When placing a PDO and choose
the supplier location, if user adds it by pressing the enter key it will not place
the PDO. If user clicks the headend with the mouse it goes right through. Issue
occurs with all browsers, including FireFox.  Asked Mark to try in UAT as well.
Said ...Just tried it in the test, same issue",
        "language": "en",
        "IT Related": "['user', 'Issue', 'FireFox']",
        "Partially IT Related": "['the enter key', 'the mouse', 'all browsers']",
        "Not IT Related": "['Customer MJOHNS17 - reports', 'a PDO', 'the supplier
location', 'it', 'the PDO', 'the headend', 'Asked Mark', 'UAT', 'the test']"
    },
```

Figure 4.5: Example 1 result with IT dictionary



Confidence: 0.5

Language: English

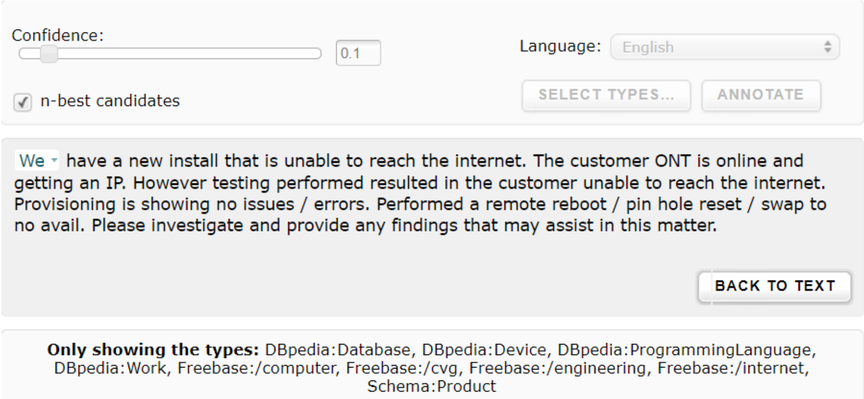
☑ n-best candidates

We have a new install that is unable to reach the internet. The customer ONT is online and getting an IP. However testing performed resulted in the customer unable to reach the internet. Provisioning is showing no issues / errors. Performed a remote reboot / pin hole reset / swap to no avail. Please investigate and provide any findings that may assist in this matter.

Only showing the types: DBpedia:Database, DBpedia:Device, DBpedia:ProgrammingLanguage, DBpedia:Work, Freebase:/computer, Freebase:/cvg, Freebase:/engineering, Freebase:/internet, Schema:Product

Figure 4.6: Example 2 result with 0.5 Confidence from DBPEDIA Spotlight

module was also able to identify '*the enter key*', '*the mouse*' and '*user*' also. In example 2, DBPEDIA Spotlight was not able to identify any concepts, however, the concept extraction module was able to detect, '*a new install*', '*the internet*', '*an IP*' and '*no issues/ error*'. From both the examples it can be seen that the proposed concept extraction module with IT domain



Confidence: 0.1

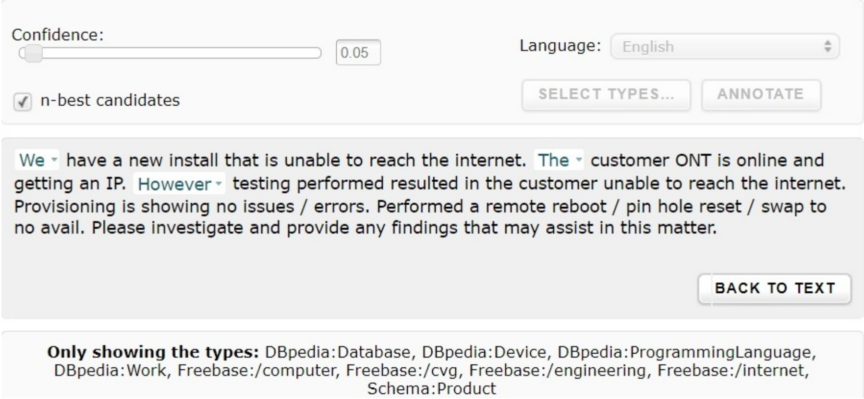
☑ n-best candidates

Language: English

We have a new install that is unable to reach the internet. The customer ONT is online and getting an IP. However testing performed resulted in the customer unable to reach the internet. Provisioning is showing no issues / errors. Performed a remote reboot / pin hole reset / swap to no avail. Please investigate and provide any findings that may assist in this matter.

Only showing the types: DBpedia:Database, DBpedia:Device, DBpedia:ProgrammingLanguage, DBpedia:Work, Freebase:/computer, Freebase:/cvg, Freebase:/engineering, Freebase:/internet, Schema:Product

Figure 4.7: Example 2 result with 0.1 Confidence from DBPEDIA Spotlight



Confidence: 0.05

☑ n-best candidates

Language: English

We have a new install that is unable to reach the internet. The customer ONT is online and getting an IP. However testing performed resulted in the customer unable to reach the internet. Provisioning is showing no issues / errors. Performed a remote reboot / pin hole reset / swap to no avail. Please investigate and provide any findings that may assist in this matter.

Only showing the types: DBpedia:Database, DBpedia:Device, DBpedia:ProgrammingLanguage, DBpedia:Work, Freebase:/computer, Freebase:/cvg, Freebase:/engineering, Freebase:/internet, Schema:Product

Figure 4.8: Example 2 result with 0.05 Confidence from DBPEDIA Spotlight

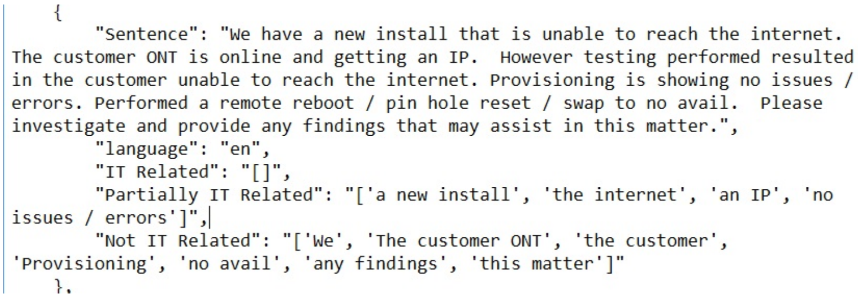

```
    {
        "Sentence": "We have a new install that is unable to reach the internet.
The customer ONT is online and getting an IP.  However testing performed resulted
in the customer unable to reach the internet. Provisioning is showing no issues /
errors. Performed a remote reboot / pin hole reset / swap to no avail.  Please
investigate and provide any findings that may assist in this matter.",
        "language": "en",
        "IT Related": "[]",
        "Partially IT Related": "['a new install', 'the internet', 'an IP', 'no
issues / errors']",
        "Not IT Related": "['We', 'The customer ONT', 'the customer',
'Provisioning', 'no avail', 'any findings', 'this matter']"
    },
```

Figure 4.9: Example 2 result with IT dictionary

specific dictionary performs better than DBpedia Spotlight for identifying IT related concepts.

Chapter 5

Conclusions and Future Work

The aim of this thesis was to develop an efficient system for extracting domain-specific information from unstructured natural language text. To achieve this goal, a comprehensive study was conducted on the sub-tasks of information extraction, namely: concept extraction and relation extraction. The study was comprised of three parts: detailed description of the tasks, a survey on related work in the recent years, and, tools and techniques that are used for accomplishing the task.

This chapter concludes this dissertation with a short summary of the developed work followed by, the limitations and future work of the proposed system.

5.1 Conclusions

In this thesis work, a working prototype of a domain-specific information extraction system was developed. The system extracts information in the form of domain-related entities and relations from plaint text in the English language and tags the sentences as the final result. The approaches used for accomplishing these tasks include dictionary matching and rule-based pattern matching. The proposed system is very similar to that of SEVA [49] in terms of approaches in achieving the goal.

The presented system was able to achieve promising results, with recall greater then 80% and precision greater then 90% for both concept and relation extraction. The system can be used for extracting structured information from unstructured text and tagging the data.

The concept extraction module displayed high performance in comparison with the available tools: Microsoft Concept Graph and DBpediaSpotlight.

5.2 Limitations

The system shows high performance in terms of precision and recall for both concept and relation extraction, however it has some major limitations as well. The foremost limitation of this system is its reliance on rule based approach for relation extraction. Manual construction of the rules are tedious and time consuming. A second key limitation is the occurrence of overlapping words between concept phrases and relational phrases. The presence of overlapping words/phrases makes the sentence tagging confusing and becoming difficult for computer to distinguish precisely between the two types of phrases. And lastly, the concept extraction module works by complete and partial dictionary match, therefore the system may be suitable for extracting biological sequences or mathematical equations from the text. It is specific for extracting phrases and words from the plain text. This limitation is assumed based on the general knowledge about the documents that contain such textual data, and the working of the proposed system.

5.3 Future Work

The proposed system can be modified in various ways, keeping it's general architecture intact.

First, the system as it is now, extracts information only from the plain text presented in English language. It can be updated to be able to extract information from many other languages as well. Two modifications are required to make the system work with multilingual text: replace the language model in the concept extraction module and replace the set of rules for verb phrase extraction.

Second, the system could be improved to avoid the occurrence of overlapping words between concept and relational phrases. This can be achieved by using various methods, like, tokenizing the relational phrases and keeping only the particular words that do not overlap, as a relational phrase.

Third, the concept extraction module only identifies entities as related or not related, but does not classify them. The system can be updated to categorize the entities. Like for Altice Labs' data the current system returns the output tagged as 'Related' and 'Not Related'. Classification the entities under categories like 'software', 'hardware', networking', etc., the system will return the output tagged as 'Software Related', 'Hardware Related', etc. Different dictionaries can be made for different categories and tags can be assigned accordingly.

Last, but not the least, with the help of the entity classification and tagged data obtained from the output processing module, various classification algorithms (like CRF) and machine learning algorithms (like language training with BERT) can be applied to make the system more effective for usage.

5.4 Ending Note

The field of information extraction is very vast, both in terms of information that needs to be extracted and its usage. New approaches, technologies and researches are emerging every year to find the best possible solutions. The theories of information extraction started sprouting from early 1960s, and after 60 years of hard work from researchers it has evolved to as we see it now. Applications like: ad-sensing, recommendation systems, automatic hate speech blocking and many more are being developed using IE, yet it would not be wrong to say that it is a long way from being complete.

With the completion of this work, I believe a small contribution was made in the evolution of information extraction. I hope the approaches used and experiments performed in this thesis will be helpful for future researches.

Appendix A

Excerpt of the Result

Query Description: When trying to enter to the production environment after loging in an error message appears (see attached). The UAT environment is OK.

Result: When {trying to enter}[Relation] to {the production environment}[Related] after {loging in}[Relation] {an error message}[Related] {appears}[Relation] ({see attached}[Relation]). The UAT environment[Related] is OK[Relation].

Query Description: network element polling executed by system pt.ptinovacao.agorang.ap.systemmonitor.SystemMonitorBean occasionally does not update internal cache or close open alarms, generating incorrect operational status in OLTs

Result: {network element polling}[Related] {executed by}[Relation] {system pt.ptinovacao.agorang.ap.systemmonitor}[Related] .SystemMonitorBean occasionally {does not update}[Relation] {internal cache}[Related] or {close open alarms}[Not Related], {generating}[Relation] {incorrect operational status}[Not Related] in {OLTs}[Not Related]

Query Discription: Hi Support, Can you please check sitecode attribute filter? It seems it isn't working. Attach there is a screenshot showing alarms for site 204, but in the top window shows not alarm. Thanks for your help. Regards, Hansell F.

Result: Hi Support, {Can}[Relation] you please {check}[Relation] {sitecode attribute filter}[Related]? It {seems}[Relation] it {isn't working}[Relation]. {Attach}[Relation] there is a {{screenshot showing}[Relation] alarms}[Related] for {site}[Related] 204, but in {the top window}[Related] {shows}[Relation] not alarm. Thanks for your help. Regards, Hansell F.

Appendix B

Rules for Verb Phrase Extraction

1. POS:VERB
2. POS:ADV, POS:VERB
3. POS:VERB, POS:ADP
4. POS:VERB, POS:ADV
5. POS:ADV, POS:ADV
6. POS:PART, POS:ADV
7. POS:AUX, (POS:PART, OP:?),POS:ADJ
8. POS:PART, (POS:AUX, OP:?), POS:VERB
9. POS:VERB, (POS:ADV, OP:?), POS:VERB
10. POS:ADJ,(POS:PART, OP:?), (POS:VERB, OP:*), POS:VERB
11. POS:VERB, (POS:DET, OP:*), (POS:ADJ, OP:*), POS:VERB
12. POS:VERB, (POS:DET, OP:?), (POS:VERB, OP:?), POS:VERB
13. POS:AUX, (POS:ADJ, OP:?), (POS:PART, OP:?), (POS:VERB, OP:*),POS:VERB
14. POS:VERB, (POS:AUX, OP:?), (POS:AUX, OP:?),(POS:PART, OP:?), POS:VERB,POS:AUX
15. POS:VERB, (POS:AUX, OP:?), (POS:AUX, OP:?), (POS:PART, OP:?), (POS:CCONJ, OP:?), POS:VERB

Here '?' makes the pattern optional, by allowing it to match 0 or 1 times and '*' allows the pattern to match zero or more times.

Bibliography

[1] Wikipedia. Information extraction.

[2] Sunita Sarawagi. Information extraction. *Fondations and Trends in Databases, Volume -1*, (3):261 – 377, 2007.

[3] Ralph Grishman. Information extraction: Techniques and challenges. *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, (2):10–27, 1997.

[4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press., 2008.

[5] Jim Cowie and Yorick Wilks. Information extraction.

[6] Peerzada Hamid Ahmad and Dr. Shilpa Dang. A comparative study on text mining techniques. *nternational Journal of Science and Research (IJSR)*, 2014.

[7] Yorick Wilks. Information extraction as a core language technology. *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, (1):1–9, 1997.

[8] Sumali Conlon, Alan S Abrahams, and Lakisha L. Simmons. Terrorism information extraction from online reports. *Journal of Computer Information Systems*, (3):20–28, 2015.

[9] Horacio Saggion, Adam Funk, Diana Maynard, and Kalina Bontcheva. Ontology-based information extraction for business intelligence. *The Semantic Web*, pages 843–856, 2007.

[10] Eduard Hovy, Nancy Ide, R. Frederking, J. Mariani, and A. Zampolli. Multilingual information management: Current levels and future abilities, 1999.

[11] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson. Fastus: A finite-state processor for information extraction from real-world text, 1993.

[12] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, and F. Feng. UMass/Hughes: Description of the CIRCUS system used for TIPSTER text. In *TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19-23, 1993*, pages 241–256. Association for Computational Linguistics, 1993.

[13] STEPHEN SODERLAND. Learning information extraction rules for semi-structured and free text. *Machine Learning, Volume -34*, page 233272, 1999.

[14] James Stuart Aitken. Laearning information extraction rules: An inductive logic programming approach. In Frank van Harmelen, editor, *ECAI 2002*, pages 355–359, 2002.

[15] Eugene Agichtein and Venkatesh Ganti. Mining reference tables for automatic text segmentation. *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004.

[16] Vinayak R. Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatically extracting structure from free text addresses. *IEEE Data Eng. Bull.*, 23(4):27–32, 2000.

[17] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Sixth Workshop on Very Large Corpora*, 1998.

[18] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Technical report, University of Pennsylvania, 2001.

[19] Soumya Ray and Mark Craven. Representing sentence structure in hidden markov models for information extraction. *IJCAI-2001*, 2001.

[20] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895. Association for Computational Linguistics, 2018.

[21] Xinbo Lv, Yi Guan, and Jinfeng Yang. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology. Volume-9*, pages 237–248, 2016.

[22] Feifan Liu, Jinying Chen, Abhyuday Jagannatha, and Hong Yu. Learning for biomedical information extraction: Methodological review of recent advances, 2016.

[23] spaCy. Models languages.

[24] NLTK. Natural language toolkit.

[25] towards data science. Traditional methods for text data.

[26] SearchBusiness analytics Margaret Rouse. unstructured text.

[27] ontotext. What is information extraction?

[28] Line Eikvil. Information extraction from world wide web - a survey, 1999.

[29] PyPi. Beautiful soup.

[30] Wikipedia. Concept.

[31] Wikipedia. Glossary of computer science.

[32] centriqs. Database entity: Definition, relationship, attributes and settings.

[33] Quora-Dave Orr. What is the difference between an entity and named entity in nlp?

[34] Claire Nédellec, Adeline Nazarenko, and Robert Bossy. *Handbook on Ontologies*, chapter Information Extraction, pages 663–685. 2009.

[35] Wikipedia. Named-entity recognition.

[36] Tome Eftimov, Barbara Koroui Seljak, and Peter Koroec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *Plos One*, 2017.

[37] Yaxi Zhang. Named entity recognition for social media text. Master's thesis, Uppsala University, 2019.

[38] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. *ANLC '97: Proceedings of the fifth conference on Applied natural language processing*, 1997.

[39] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy markov models for information extraction and segmentation. *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, page 591598, 2000.

[40] Koichi Takeuchi and Nigel Collier. Use of support vector machines in extended named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[41] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *CONLL '03: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, page 188191, 2003.

[42] Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. A decision tree method for finding and classifying names in Japanese texts. In *Sixth Workshop on Very Large Corpora*, 1998.

[43] Wenhui Liao and Sriharsha Veeramachaneni. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 58–65. Association for Computational Linguistics, 2009.

[44] Shubhanshu Mishra and Jana Diesner. Semi-supervised named entity recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212. The COLING 2016 Organizing Committee, 2016.

[45] David Nadeau, Peter D. Turney, and Stan Matwin. Ai'06: Proceedings of the 19th international conference on advances in artificial intelligence: Canadian society for computational studies of intelligence. *CONLL '03: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, page 266277, 2006.

[46] Dekang Lin and Patrick Pantel. Induction of semantic classes from natural language text. *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, page 317322, 2001.

[47] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles", booktitle = COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics. pages 848–853. COLING, 2004.

[48] Wikipedia. Insideoutsidebeginning (tagging).

[49] Jitin Krishnan, Patrick Coronado, Hemant Purohit, and Huzefa Rangwala. Common-knowledge concept recognition for seva, 2020.

[50] Hong-Woo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In Russ B. Altman, Tiffany Murray, Teri E. Klein, A. Keith Dunker, and Lawrence Hunter, editors, *Biocomputing 2006, Proceedings of the Pacific Symposium, Maui, Hawaii, USA, 3-7 January 2006*, pages 4–15. World Scientific, 2006.

[51] Alexandra Pomares Quimbaya, Alejandro Sierra, Rafael A Gonzalez, Julian Daza, Oscar Mauricio Muñoz, and Angel A. García. Named entity recognition over electronic health records through a combined dictionary-based approach. *Conference: International Conference on ENTERprise Information Systems/International Conference on Project MANagement/International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist. Volume -100*, 2016.

[52] K.R. Rahem and Nazlia Omar. Rule-based named entity recognition for drug- related crime news documents. *Journal of Theoretical and Applied Information Technology. Volume -77*, (2):229–235, 2015.

[53] Raghu Anantharangachar, Srinivasan Ramani, and S. Rajagopalan. Ontology guided information extraction from unstructured text. *International Journal of Web Semantic Technology (IJWesT) Volume -4*, 2013.

[54] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology - Volume 9*, pages 237–248, 2016.

[55] Zhongyuan Wang and Haixun Wang. Understanding short texts. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts.* Association for Computational Linguistics, 2016.

[56] Microsoft. Microsoft concept graph for short text understanding.

[57] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. Probase: A probabilistic taxonomy for text understanding. *SIGMOD '12: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, page 481492, 2012.

[58] Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. An inference approach to basic level of categorization. *CIKM '15: Pro-*

ceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015.

[59] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. Query understanding through knowledge-based conceptualization. *IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence*, 2015.

[60] Z. Wang, H. Wang, and Z. Hu. Head, modifier, and constraint detection in short texts. In *2014 IEEE 30th International Conference on Data Engineering*, pages 280–291, 2014.

[61] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2330–2336. IJCAI/AAAI, 2011.

[62] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou. Short text understanding through lexical-semantic analysis. In *2015 IEEE 31st International Conference on Data Engineering*, pages 495–506, 2015.

[63] DBpedia. Dbpedia spotlight shedding light on the web of documents.

[64] cortical.io. cortical.io.

[65] National Library of Medicine. Unified medical language system (umls).

[66] cortical.io. cortical.io in a nutshell.

[67] MonkeyLearn. How monkeylearn works.

[68] IBM. Watson natural language understanding.

[69] TAmazon web Services (aws). Amazon comprehend discover insights and relationships in text.

[70] TextRazor. Extract meaning from your text.

[71] Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[72] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[73] Wikipedia. Hyponymy and hypernymy.

[74] Jackson M. Steinkamp, Charles Chambers, Darco Lalevic, Hanna M. Zafar, and Tessa S. Cook. Toward complete structured information extraction from radiology reports using machine learning. *Journal of Digital Imaging Volume -32*, 2019.

[75] Trapit Bansal, Pat Verga, Neha Choudhary, and Andrew McCallum. Simultaneously linking entities and extracting relations from biomedical text without mention-level supervision. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(05)*, pages 7407–7414, 2020.

[76] Yue Zhao and John Handley. Extracting clinical concepts from user queries, 2019.

[77] ALISA SMIRNOVA and PHILIPPE CUDRÉ-MAUROUX. Relation extraction using distant supervision: a survey. *ACM Computing Surveys - Volume 51*, (5), 2018.

[78] Adam Grycner and Gerhard Weikum. POLY: Mining relational paraphrases from multilingual sentences. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2183–2192. Association for Computational Linguistics, 2016.

[79] Kamel Nebhi. A rule-based relation extraction system using dbpedia and syntactic parsing. *NLP-DBPEDIA'13: Proceedings of the 2013th International Conference on NLP DBpedia - Volume 1064*, page 7479, 2013.

[80] Mujiono Sadikin. A novel rule based approach for entity relations extraction. *Journal of Theoretical and Applied Information Technology*, 2015.

[81] Isabel Segura-Bedmar, Paloma Martínez, and César de Pablo-Sánchez. A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. *BMC Bioinformatics 12 Suppl 2(Suppl 2):S1*, 2011.

[82] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *ACLdemo '04: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22es, 2006.

[83] Shantanu Kumar. A survey of deep learning methods for relation extraction, 2017.

[84] Nguyen Bach and Sameer Badaskar. A review of relation extraction.

[85] Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. Relation extraction : A survey, 2017.

[86] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: the second generation. *IJCAI'11: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume 1*, page 310, 2011.

[87] Andreas Herman. Different ways of doing relation extraction from text, 2019.

[88] K.E. Ravikumar, Majid Rastegar-Mojarad, and Hongfang Liu. Belminer: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. *Database. Volume 2017*, 2017.

[89] Peng Xu and Denilson Barbosa. Connecting language and knowledge with heterogeneous representations for neural relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3201–3206. Association for Computational Linguistics, 2019.

[90] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905. Association for Computational Linguistics, 2019.

[91] Evan Sandhaus. The new york times annotated corpus, 2008.

[92] The National Aeronautics and Space Administration. Stardust/next.

[93] DBpedia. Dbpedia.

[94] Python. Python.

[95] Wikipedia. Python (programming language).

[96] PyPi. langdetect 1.0.8.

[97] PyPi. pandas 1.1.4.

[98] spaCy. Industrial-strength natural language processing in python.

[99] SPARQL. Sparql endpoint interface to python.

[100] Wikipedia. Verb phrase.

[101] British Council. Verb phrases.

[102] spaCy. Matcher.

[103] spaCy. Matcher – matcher.add.

[104] Python. re regular expression operations.

[105] DP solutions problems solved. Technology terms.

[106] DATAPRISE. It glossary.

[107] ourcommunity.com.au. The a-z of technology terms.

[108] Angelo State University. Information technology glossary.

[109] The University of Utah. Online computer science glossary.

[110] Quick Base. Glossary of computer related terms.