A Study on Various Text Document Annotation Systems

Sajith H, Vipin Kumar KS

Abstract— **Annotations are the tags, keywords, attribute-value pairs, comments or summary that are attached to a document or a part of the document. In case of text documents, these annotations are generated mostly based on the content of the document. Annotations are necessary for effective searching and retrieval of the documents. The concept of semantic web is closely related to annotation of documents. With a lot of research being carried out in this area, annotation systems have evolved from manual to semi-automated and fully automated systems. Researchers are also concerned about improving the performance of annotation systems in terms of accuracy of the annotations generated. This paper examines the various text based document annotation tools and frameworks and gives a detailed comparison about them.**

Index Terms— **Annotation, Metadata, Semantic web, Text document.**

I. INTRODUCTION

In the past few years, there had been an explosion of data in the World Wide Web. There are many data or document sharing platforms like news blogs, social networking sites, etc. through which huge amount of data are uploaded and shared on a daily basis. Effective handling of this data is thus necessary for searching and retrieving the documents or information present in the documents efficiently. The increased necessity for intelligent knowledge management has led to the advent of semantic web [1]. The semantic web which is an extension of the existing web provides an easier way for sharing, searching and retrieval of information from the web. The basic ideology of semantic web is to add semantic metadata or annotations to the documents or part of a document. A lot of research is being carried in this area recently. As a result of which the metadata or annotation generation systems have evolved from manual to semi-automatic and to fully automatic.

Annotations gives a structured form for the information present in the documents. It can be in the form of tags, keywords, attribute-value pairs, comments or summary. Semantic annotation deals with the content (textual data in the text document) of the document. Semantic or content based annotations are applicable not only to text documents but also to other digital forms of documents such as images, audio and video. In this paper, we focus on annotation of text documents. Various features and concepts are to be considered in the field of annotation generation of digital contents. A formal model of annotation of digital contents is proposed in [2]. The aim of the formal model is to formalize the main concepts concerning annotations and to define the relationships between annotations and annotated information resources. The model covers areas like Identification, Cooperation, Semantics, Linking, and Materialization.

Besides enhancing document searching and retrieval, the annotation of documents also helps in obtaining structured information from unstructured collection of documents. For example, consider a website which reviews mobile phone models. The review about a particular model will be written in the form of an article arranged using paragraphs. By semantic annotation, only the most important information about this particular model is made available in a structured format. This could be the specifications of the model which may include its display size, resolution, camera features, etc. Hence a person who only want to know the specifications of a mobile phone doesn't have to read the entire review about the phone.

This paper aims at providing a detailed study of various text document annotation tools and frameworks. The rest of this paper is organized as follows. Section 2 describes some of the important concepts related to text based document annotation and section 3 describes the classification of annotation systems. Section 4 explains various document annotation tools and frameworks. Finally section 5 gives a comparison of these tools and frameworks.

II. IMPORTANT CONCEPTS

1. Semantic web

The semantic web can be considered as a web of data. It is an extension of the existing web which aims to extend the principles of web from documents to data. It provides a common framework for finding, sharing, reusing and combining information present in the web document. It is a collaborative effort led by the World Wide Web Consortium (W3C). The basic working principle of semantic web is to add semantic metadata or annotation to web documents. The concepts of semantic web and Big data are closely related [3]. As web grows at a high rate resulting in Big data, annotating the web documents becomes a challenging task. Manual annotation become quite impossible in such a scenario.

Manuscript received June, 2015.

***Sajith H**, M.Tech Student, Computer Science and Engineering Department, Government Engineering College, Thrissur, India*

***Vipin Kumar KS**, Asst. Professor, Computer Science and Engineering Department, Government Engineering College, Thrissur, India*

Automatic annotation techniques are hence necessary to make the idea of semantic web possible.

2. Ontology

An ontology is a formal model for representation of objects, their properties and interrelationships in a particular area or domain of interest [4]. It can be considered as a specification of conceptualization. Most ontologies describe individuals (instances), classes (concepts), attributes and relations. Individuals includes the instances, objects or entities. Classes are collections, concepts, types of objects or kind of things. Attributes includes the features or properties or characteristics that the objects or classes can have and the relations are the ways in which these classes and individuals can be related to one another.

3. Knowledge base

A knowledge base is a database which can be considered as a repository of information and this information can be accessed and manipulated in some predefined fashion. The knowledge present in a knowledge base can be said to be modeled according to an ontology. That is, an ontology is a representation of the knowledge base.

III. TYPES OF ANNOTATION SYSTEMS

Based on the human intervention factor, annotation systems can be classified in to three types: Manual annotation systems, Semi-automatic annotation systems and automatic annotation systems [5].

1. Manual Annotation

Manual Annotation deals with adding metadata tags or keywords for a document or part of a document manually by the user. Either the author who created the document or others who later uses the document can add annotations. Manual annotations are the oldest form for adding annotations to a document. When the number documents that need to be annotated is very high, manual annotation become time consuming or practically impossible.

2. Automatic annotation

Automatic annotation is the process of adding annotations to a document or part of a document using an annotation tool or knowledge extraction tool without the help of human users. Many such tools have been implemented recently to generate automatic annotations for a document. An important factor to consider here is the accuracy of the annotations generated.

3. Semi-Automatic annotation

Semi-automatic annotation systems are also automatic annotation tools or mechanisms but they also involve some form of human intervention. They usually make their annotation suggestions and user can approve or disprove those suggestion. For example some systems allow users to inspect the annotations generated by the system and allow them to edit them if needed to improve accuracy.

IV. TEXT DOCUMENT ANNOTATION SYSTEMS

1. PANKOW

PANKOW (Pattern based annotation through knowledge on the web) [6] is an annotation method which employs an unsupervised pattern based approach. It works by categorizing instances (candidate noun phrases) with regard to a given ontology. It is called unsupervised because it does not rely on any training data annotated by hand. It is called pattern-based because it makes use of linguistically motivated regular expressions in order for identifying the instance-concept relations in the text. It is implemented in Ont-O-Mat which is an annotation tool used for the semantic web. The annotation process using PANKOW consists of four steps. First a web page is given as the input. The POS tagger in the system scans the webpage for candidate phrases that can be categorized as instances of the ontology. Candidate phrases are proper nouns. The system then derives hypothesis phrases by using candidate proper nouns and all candidate ontology concepts. Then Google is queried for the hypothesis phrases through its web service API. For each hypothesis phrase, number of documents that contain it is found out. As a result, we get the number of hits for each hypothesis phrase. For each instance-concept pair, the total sum of the query result is calculated. The candidate proper nouns are then categorized into their highest ranked concepts. Hence it annotates a piece of text as describing an instance of that concept. Thus we get an ontologically annotated web page. Three methods are used for categorizing candidate noun phrases namely, base line, linear weighting and interactive selection. PANKOW is integrated in to the CREAM framework. It extends the CREAM implementation Ont-O-Mat. Ont-O-Mat supports two modes of Interaction with PANKOW. One is fully automatic and the other is Interactive semi-automatic. An enhanced version of version of PANKOW called C-PANKOW (Context driven PANKOW) is proposed in [7].

2. MUSE

MUSE [8] is an information extraction system developed with in the GATE architecture. MUSE performs named entity (NE) recognition and coreference on a variety of different types of text, such as news reports, emails, and spoken transcriptions. MUSE System consist of six components. They are Tokeniser, Sentence Splitter, POS Tagger, Gazetter Lists, Semantic Tagger and Orthomatcher. Tokeniser is used for splitting the text into simple tokens such as numbers, punctuations, words of different type and white space. Sentence splitter segments the running text into sentences. It is implemented in JAPE (Java annotation pattern engine). POS tagger produces a POS tag (Eg: Verb, noun, etc.) as an annotation on each token. It uses a default lexicon and rule set, created by training on an annotated. Gazetter list is used to help Named entity recognition. Each list represent a set of names, such as Cities, Organizations, days of week, first names, etc. and an index file is used to access them. Such lists are not sufficient because such names can be highly ambiguous (Eg: India could be name of a person or country) and also it is impossible to create such exhaustive lists. Semantic tagger is implemented using JAPE which use previously generated annotations to recognize Name entities. Semantic tagger recognize annotation from POS tagger and gazetteer and combine them to produce new Named entity annotations over patterns. Orthomatcher performs orthographic coreference between named entities in the text by using a set of hand crafted rules.

3. GoNTogle

GoNTogle [9, 10] is a tool which provides advanced document annotation and search facilities. The main features of GoNTogle are the following. It supports several widely used document formats such as .pdf, .doc, etc. It provides an easy and intuitive way of annotating documents (or document parts) using OWL and RDF/S ontologies. Its annotation mechanism is based on models trained from user annotation history and hence the annotation favors user interest. It provides both manual and automatic annotation for whole or part of the document. The automatic annotation mechanism is based on weighted kNN classification concept that exploits user annotation history to automatically suggest annotations. It provides a collaborative environment for users to annotate and search documents as it uses a server based architecture and document annotations are stored in a central repository. It combines keyword and semantic based search. It also offers advanced ontology query facilities. The GoNTogle system consists of 4 components. The semantic annotation component provides facilities regarding the semantic annotation of documents. It consists of three modules namely Document Viewer, Ontology Viewer and Annotation Editor. Ontology Server Component stores the semantic annotations of documents in the form of class instances. It consists of two modules, an Ontology Manager and an Ontology Knowledge Base.

4. KIM

The KIM [11] platform enables automatic semantic annotation, indexing and retrieval of documents. Named entities mentioned in the documents constitute an important part of their semantics. Semantic annotations in KIM is based on this hypothesis. Named entities means people, organizations, locations and others referred by name. Semantic annotation is done by assigning links between entities in the text and their semantic descriptions. The information extraction in KIM is performed based on an ontology and a massive knowledge base. The ontology used by KIM is called KIM ontology (KIMO). The KIM ontology contains definitions of entity classes, attributes, and relations, and also a branch of lexical resource types. The knowledge base is used for keeping semantic descriptions of entities and relations. For each entity reference in the text, KIM provides two links. One is a link to the most specific class in the ontology and another is a link to the specific instance in the KB. KIM information extraction is based on GATE framework. GATE is a widely used framework and graphical development environment. It enables users to develop and deploy language engineering components and resources in a robust fashion. Most of GATE's document management functionality and generic NLP components like Tokenizer, Part-of-Speech Tagger, and Sentence Splitter are used in KIM. Changes are made in the grammar rules. The grammar rules are based on the ontology classes, instead of on a flat set of NE types. KIM also allows the annotations to be indexed with respect to the named entities. This helps in the future document retrievals. While searching, users specify the named entities that need to be referred in the documents of interest, with name restrictions. Further, pattern of entities, relations between them and attribute restrictions can be specified. Semantic restrictions over the entities in the Knowledge base are used to answer the queries.

5. Onto-O-Mat

Onto-O-Mat [12] is an annotation tool that is implemented using S-CREAM (Semiautomatic CREAtion of Metadata) framework. S-CREAM allows creation of metadata and it can be trained for a specific domain. It supports semi-automatic annotation of webpages. Ont-O-Mat performs semi-automatic annotation by using the information extraction component Amilcare. Amilcare is a system that learns information extraction rules from manually marked-up inputs, or in other words, manually annotated documents. Amilcare uses the ANNIE (A Nearly-New IE system) which is a part of the GATE toolkit in order to perform information extraction. User annotations are provided as XML tags to train Amilcare's learner. It induces rules that are able to reproduce the text annotation. Amilcare can work in two modes. The training mode is used to adapt to a new application whereas the extraction mode is used to actually annotate the text. In both modes, the first step Amilcare do is preprocessing the texts using ANNIE. ANNIE performs tokenization (segmenting texts into words), sentence splitting (identifying sentences), part of speech tagging (lexical disambiguation), gazetteer lookup (dictionary lookup), and named entity recognition (recognition of people and organization names, dates, etc.). Amilcare induces rules for information extraction in the training mode. The learner is based on Lazy-NLP covering algorithm for supervised learning of information extraction rules. A collection of rules for Information extraction that are associated to the specific scenario is obtained as the output of training phase. In extraction mode, Amilcare receives a text or a collection of texts with the associated scenario as input. It preprocesses the text by using ANNIE and then it applies its rules and returns the original text with the added annotations.

6. SemTag

SemTag [13] is an automated semantic tagging application written on the Seeker platform. It can perform semantic tagging of large corpora. Seeker is a platform which supports large-scale text analytics. SemTag performs the annotation process in three passes namely Spotting pass, Learning pass, and Tagging pass. Spotting pass includes retrieving and tokenizing the documents from the seeker store. They are then processed to find label matches from TAP taxonomy. The labels that match are saved with a ten word window to either side of the particular candidate object. The learning pass includes scanning a representative sample of the corpus in order to find the corpus-wide distribution of terms at every node of the taxonomy. Finally, in the tagging pass, the windows are scanned and the matches are disambiguated. When a label and an actual TAP object are found matching, then the URL, the reference and other metadata are entered into the database as final results. SemTag uses TAP taxonomy which is a continuously evolving ontology. SemTag uses Taxonomy based disambiguation algorithm (TBD) for resolving ambiguities in the corpus.

7. TEXTRUNNER

TEXTRUNNER [14] is a fully implemented and highly scalable Open Information extraction (OIE) system. Open Information Extraction is an extraction paradigm which enables domain independent discovery of relations extracted from text. It can readily scale to the diversity and size of the Web corpus. When a corpus is given as the input to an OIE system, a set of extracted relations are returned as output. TEXTRUNNER is the first domain-independent OIE system. It works by linguistically parsing the natural language sentences. The results after parsing are then used to obtain several candidate tuple extractions. Finally, the accuracy of the extractions are determined by using extraction frequency based. TEXTRUNNER consists of three modules. First is a self-supervised learner module. In this module, the candidate extractions are labelled as trustworthy or not. The labelling is done based on a small corpus sample given as input. Next is a single pass extractor module in which a single pass is made over the entire corpus. Each word in each sentence is automatically tagged with the most probable part-of-speech in this module. These tags are used for finding entities and relations. A lightweight noun phrase chunker is used for finding out the entities. Relations are found out by eliminating non-essential phrases heuristically from the text between noun phrases. The final module consist of a redundancy based assessor which assigns a probability to each of the retained tuple. It is done based on a probabilistic model of redundancy.

8. AeroDAML

AeroDAML [15] is a knowledge markup tool used for generating DAML annotations. The DARPA Agent Markup Language (DAML) is a markup language which is developed as an extension of XML and RDF. DAML annotation of documents and web pages is a complex and time consuming task. In AeroDAML, natural language information extraction techniques are applied to generate DAML annotations automatically from web pages. The ontology used by AeroDAML is a default generic ontology which consist of commonly found word classes and relationships. When the user enters the URI of webpage, AeroDAML generates the DAML annotations for that particular webpage. There is a client server version of AeroDAML and it enables annotation using customized ontologies. In client server version, the user enters a file name instead of URI and AeroDAML generates the DAML annotation for that text document. A commercial information extraction product called AeroTextTM is used by AeroDAML for generating annotations. AeroTextTM is a high performance, versatile information extraction system which provides advanced graphical tools and supports a variety of text processing tasks. It is composed of a Knowledge Base Compiler and a Knowledge Base Engine. The Knowledge Base Compiler is used for converting the linguistic data files into a knowledge base (KB). The knowledge base is applied to input documents by the Knowledge Base Engine. AeroTextTM also contains an Integrated Development Environment (IDE) and a Common Knowledge Base. The IDE provides the environment for handling linguistic knowledge bases. The Common Knowledge Base is the component which enables extraction of most proper nouns and frequently occurring relations using domain independent rules.

9. OnTea

OnTea (Ontology based text annotation) [16] tool is a semi-automatic ontology based text annotation tool was created as a part of the NAZOU project. In OnTea, text or a text document is analyzed using regular expression patterns. Equivalent semantic elements are detected according to the defined domain ontology. The input to OnTea are text resources like HTML, email or plain text. New ontology individuals corresponding to the annotated text are generated as output. The detected ontology individuals are then used to fill the properties of this new individual according to defined patterns. OnTea uses RDF/OWL ontologies. The implementations are carried out in Java using the Jena Semantic Web Library and Sesame Library.

10. PIRATES

PIRATES [17] or Personalized Intelligent Tag Recommender and Annotation TEStbed is a framework for text based content retrieval, annotation and classification. It uses an unsupervised approach to recommend significant metadata for a given web document. It combines tags, keyphrase extraction, and ontology mining and assists the user when he/she tags a web resource. In order for keyphrase extraction, PIRATES introduces features that combine linguistic knowledge with the statistical features. PIRATES includes an unsupervised domain independent algorithm which works without a specific domain model and a prior knowledge about the nature of the document set. The domain independent extraction algorithm is called DIKpE (Domain

Independent Keyphrase Extraction) algorithm. There are mainly three steps in keyphrase extraction. First is extracting candidate phrases from document. This is done through Format conversion, cleaning and sentence delimiting, POS tagging and n-gram extraction, stemming and stop word removing, and separating n-gram lists. Next is feature calculation which is done using the five features which are phrase frequency, POS value, phrase depth, phrase last occurrence and phrase lifespan. The final step is scoring and ranking according to keyphraseness.

11. Context Aware Search

In context aware search [18], a semantic information retrieval technique using ontology is used. It is based on the idea that maintaining a dynamic and evolving domain ontology in order to accommodate retrieved information can improve the precision of retrieval process. Searching is performed by interpreting the meanings of keywords provided by domain ontology. Ontology together with instance of the class constitutes a knowledge base. Information contained in the digital documents are extracted and stored in Jena based triple store. The architecture of the system consist of three modules. They are Knowledge extractor, Ontology change management and search module. In Knowledge extractor module, semantically aware metadata of a document is generated. In this module steps like transforming the document into standard format, component identification, term extraction, lexical, hierarchy identification, knowledge representation and knowledge verification are performed. Ontology change management module deals with modifying or updating the ontology according to the changes in the domain knowledge. Ontology enrichment and ontology population are the two basic operations performed in this module. Changes are detected using H-Match Algorithm and the changes are represented using Change History Ontology (CHO). A change history log (CHL) is used to keep track of all changes made to the domain ontology. Document searching module deals with submission of queries and retrieval of relevant documents.

12. Search Result Annotation

In search result annotation [19], a multi annotator approach is used to automatically annotate search results from web databases. Annotations are performed at the data unit level. Annotations or labels are assigned to the data units with in the search result record (SRR) returned from WDBs. Each SRR returned from a WDB contain multiple data units. This annotation approach consist of three phases. The alignment phase begins with identifying all the data units in the SRRs and then organizing them in to different groups. Each such group corresponds to a different concept. Data alignment helps in finding common patterns and features among data units and makes annotation mechanism easier to perform. Whether data units belong to same concept are determined by data content, presentation style, data type, tag path and adjacency. The annotation phase consist of annotation performed by six basic annotators. They are Table annotator (TA), Query-Based Annotator (QA), Schema Value Annotator (SA), Frequency Based Annotator (FA), In-Text Prefix/Suffix Annotator (IA) and Common Knowledge Annotator (CA). A probabilistic method is adopted to combine these annotators. Annotation wrapper phase consist of constructing an annotation wrapper for the WDB from the annotated data units. This wrapper can be used to annotate new SRRs easily without reapplying the entire process.

13. Artequakt Project

Artequakt project [20] aims at implementing a system for extracting knowledge about artists from web and populate a knowledge base. This knowledge is then used to generate personalized narrative biography of the artists. Artequakt project makes use of three existing projects namely, Artiste project, The Equator IRC and AKT IRC. The ontology used by Artequakt project is implemented in protégé. The ontology represents the domain of artists and artefacts. For knowledge extraction, WordNet lexical database and GATE entity recognizer are used as guidance tools and they helps in identifying entities and relations between knowledge fragments. The Knowledge extraction procedure in Artequakt project is as follows. First the user enters an artist name and the system makes a quick search in knowledge base to check if it already exists. If the given artist is new to knowledge base, it is searched in web using Yahoo and Altavista search engines and a selection of relevant documents is made. Then each of these documents are divided into paragraphs and then into sentences. Syntactic and semantic analysis of every paragraph is carried out in order for finding relevant knowledge to extract. The grammatically related phrases generated as a result of the syntactic analysis are grouped by using Apple pie parser. Named entities and the binary relationships are identified then by using GATE and WordNet. These information are stored in the knowledge base and Narrative construction tools are used to generate biographies of the artists.

14. NATM

NATM (Noisy annotated topic model) [21] is a probabilistic topic model which works on noisy annotated data. It analyses and extracts content related annotations from noisy annotated data. Noisy annotations are content unrelated annotations. In a topic model, documents are modeled as mixture of topics and topics are modeled as probability distribution of words. NATM is an extension of Corr-LDA (Correspondence latent Dirichlet allocation). NATM enables the content related annotation feature which is not supported by Corr-LDA. NATM uses an unsupervised approach in which annotations are extracted without content relevance tables. NATM can be considered as a generative model for content and annotation. Contents are generated first followed by annotations. Annotation generation depends on a latent variable. The latent variable indicates whether the annotations are related to the content or not. Annotations are generated either from content generating topics or from content unrelated general distribution by using the inference model. The inference of the latent topics are computed using collapsed Gibbs sampling.

15. E-Learning Annotation

In E-Learning annotation [22], a model that extends the IEEE LOM (Learning Object Metadata) standard with ontology-based semantic annotations is introduced. This model makes use of existing approaches that adopt ontologies in order for annotating e-learning resources. Semantic annotation of the e-learning materials makes use of a manual annotation component and semi-automatically annotation component. The manual annotation component is used mainly for the LOs in image/audio/video format. Semi-automatically annotation component deals with textual LO. It is implemented by combining semantic technologies with natural language processing techniques. It makes use of TFxIDF indexing, latent semantic indexing and word net based processing for annotation of textual LO. The first step is document preprocessing in which documents are loaded into memory and then they are split into tokens. Porter stemmer is used for stemming the tokens. Second step involves the calculation of a frequency table which contains the document's tokens along with their frequencies sorted in descending order. The Term Frequency (TF) matrix is created which have terms or tokens as rows and documents as columns. The next step is obtaining the TFxIDF matrix from the TF matrix. It is done by using Inverse Document Frequency Indexing. After TFxIDF matrix is computed, the Singular Value Decomposition method is applied to this matrix to reduce its dimensionality and also reveals latent relationships among documents based on word co-occurrences. The next step which is concept matrix construction is done by using Latent Semantic Indexing technique. The concept matrix is then reduced and finally the result matrix is obtained which is then evaluated.

16. Deep Annotation

Deep annotation [23] deals with annotation of dynamic web pages. Deep annotation is the process of mapping the information or information structure or information context to other information structures. These mappings can be used to query the database for retrieving semantic data from the website. Deep annotation process consist of four steps. In the first step, server side markup is created by the database owner. Then the client side annotations are produced by the annotator. The mapping rules that are derived from the annotation along with client ontology is published by the annotator. Finally the ontology and mapping rules are used by the querying party to query the database. Database owner, annotator and querying party can be considered as the three pillars of deep web annotation architecture. Relational metadata is created by using an extended version of OntoMat-Annotizer. OntoEdit is used for investigating, debugging and changing the mapping rules.

17. SHOE Knowledge Annotator

SHOE knowledge annotator [24] is an annotation tool that allows users to markup webpages with SHOE (Simple HTML Ontology Extensions) ontology. It aims at providing users with tools that allow them to create markups by making selections and filling forms. The user will be provided with an interface where he can add, edit or remove instances or ontologies. Shoe knowledge annotator checks for correctness of the markups and converts them to legal SHOE syntax. Running SHOE is an enhanced version which enables automatic markup of webpages by specifying a series of delimiters and creating a wrapper. The tool displays a table which contains a row for each record and a column for each field. This table is then converted into SHOE markup. The tool also lets the user to specify a series of templates which helps in classification and relation declarations. The use of templates enables easy regeneration of SHOE markup in case the content of the page changes.

18. MnM

MnM [25] is an annotation tool that provides both automatic and semi-automatic annotation features for annotating webpages. MnM is a combination of ontology servers, information extraction tools and augmented web browsers. It works by integrating the web browser with an ontology editor and also provides open APIs for integrating information extraction tools and for linking to ontology servers. The working of MnM includes mainly five activities namely browse, markup, learn, test and extraction. A library of knowledge models from the web are browsed and a specific knowledge set is selected by the user. A hand-crafted KMi ontology is used for annotating the documents with a set of tags. Learning phase makes use of Amilcare and Annie for enabling information extraction through learning of extraction rules in the form of tagging rules and correction rules. A test corpus and training data is used for testing the performance of the information extraction mechanism. Finally the induced rules are used for extracting information from texts.

19. CADS

CADS or Collaborative adaptive data sharing platform [26, 27] is a content based document annotation and retrieval system. It generate annotations in the form of attribute-value pairs. It relies on the idea that humans are likely to add the necessary annotations while uploading the document. CADS system automatically generates attributes based on the content of the document and the query workload. Two parameters namely, content value and querying value are used to obtain a score for each attribute and the top-k attributes are generated based on this score, where k can be any predetermined value. Two approaches, one based on the

Bayes model and the other based on Bernoulli model are used to calculate the score. Since, the query workload and previous annotations are used to generate new annotations, the CADS facilitates annotation generation in accordance to the user interest.

V. COMPARISON

Table I shows the comparison of various annotation tools and frameworks based on their approaches and implementation details like document formats, tools used, etc.

Table I: Annotation Tools Comparison

Topic	**Approach**	**Tools or algorithms used**	**Ontology**	**Standard format**
PANKOW [6]	Unsupervised, pattern based annotation	Qtag and Tree Tagger	Given ontology	HTML
MUSE [8]	Rule based Named entity recognition and coreferencing	JAPE and Brill Tagger	Given ontology	XML, HTML, SGML, email, etc.
GoNTogle [9]	Supervised, Rule based annotation	Lucene, Protégé Server & MySQL Server, Open Office API and Multivalent	OWL & RDF/S ontologies	PDF, .DOC, TXT, RTF, ODF, SXW, etc.
KIM [11]	Rule based named entity recognition	Sesame RDF Repository, Lucene IR Engine	Prebuilt KIM Ontology	HTML
Ont-O-Mat [12]	Rule based wrapper induction	Amilcare, ANNIE	Given ontology	HTML
SemTag [13]	Supervised pattern matching	TBD algorithm	TAP Ontology	HTML
TEXTRUNNER [14]	Rule based Open information extraction	Naïve Bayes Classifier, noun phrase chunker	-	HTML
AeroDML [15]	Pattern based DAML Annotation	AeroText^{TM} Java API	WordNet & AeroText knowledge base	DAML
OnTea [16]	Ontology based Pattern matching	Java Semantic Web Library or Sesame Library	RDF/OWL ontology	HTML, Plain text, email
PIRATES [17]	Unsupervised, domain independent tag generation	Stanford POS tagger, Porter Stemmer algorithm	OWL ontology	Any text document format
Context aware search [18]	Dynamic domain ontology based semantic information retrieval	OntoWordNet, SemRef and SemEVal	Rhetoric structure ontology and change history ontology	Any text document format
Search Result Annotation [19]	Rule based, wrapper based multi annotation	CombMNZ algorithm	-	Search result records (SRR)
Artequakt project [20]	Named entity recognition and binary relationship identification	Protégé, WordNet and Gate entity recognition tools	Artequakt ontology (Constructed form CRM ontology)	HTML
NATM [21]	Inference based annotation topic model	Collapsed Gibbs sampling	-	Any text document format
E-Learning Annotation [22]	latent semantic indexing and word net based processing	Porter Stemmer	Learning object ontology	IEEE Learning objects
Deep web [23]	Dynamic webpage annotation	OntoMat-Annotizer, OntoEdit, Ontobroker Inference engine	-	HTML
SHOE knowledge annotator [24]	Ontology based webpage markup generation	Expose	Given ontology	HTML
MnM [25]	Ontology based wrapper induction	Amilcare, ANNIE, Lazy NLP algorithm	KMi ontology	HTML

CADS [26]	Attribute-value pair annotation using query workload	Bayes and Bernoulli equations	-	Any text document format

The performance of annotation tools are compared on the basis of precision, recall and f-measure. Precision implies how many of the generated annotations are relevant and recall implies how many relevant annotations are generated. F-Measure is the harmonic mean of precision and recall. Table II shows the comparison of some of the annotation tools.

Table II Performance comparison

SL No.	**Topic**	**Precision %**	**Recall %**	**F-Measure %**
1	PANKOW [6]	65	28	25
2	MUSE [8]	94	92	93
3	GoNTogle [9]	80	90	-
4	KIM [10]	86	82	84
5	SemTag [13]	82	-	-
6	TEXTRUNNER [14]	88	-	-
7	OnTea [16]	63	83	70
8	Search Result Annotation [19]	97	98	-

VI. CONCLUSION

In this paper, details of various text document annotations tools and frameworks have been presented. There are mainly three types of annotation systems. They are manual, automatic and semi-automatic annotation systems. The performance of these annotation systems are compared based on three parameters namely precision, recall and f-measure. A lot of research is going on in this field to improve the performance of automatic annotation systems. A key consideration of these research is in improving the accuracy of relevant annotations being generated.

REFERENCES

[1] Berners-Lee, Tim, James Hendler, and Ora Lassila, "The semantic web." Scientific american 284.5 (2001): 28-37.

[2] Agosti, Maristella, and Nicola Ferro, "A formal model of annotations of digital content." ACM Transactions on Information Systems (TOIS) 26.1 (2007): 3.

[3] Wu, Xindong, "Data mining with big data." Knowledge and Data Engineering, IEEE Transactions on 26.1 (2014): 97-107.

[4] Breitman, Karen Koogan, and J. C. Sampaio do Prado Leite, "Ontology as a requirements engineering product." Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International. IEEE, 2003.

[5] Erdmann, Michael, "From manual to semi-automatic semantic annotation: About ontology-based text annotation tools." Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content. Association for Computational Linguistics, 2000.

[6] Cimiano, Philipp, Siegfried Handschuh, and Steven Staab, "Towards the self-annotating web." Proceedings of the 13th international conference on World Wide Web. ACM, 2004.

[7] Cimiano, Philipp, Günter Ladwig, and Steffen Staab, "Gimme the context: context-driven automatic semantic annotation with C-PANKOW." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.

[8] Maynard, Diana, "Multi-source and multilingual information extraction," Expert Update 6, No. 3 (2003): 11-16.

[9] Giannopoulos, Giorgos, Nikos Bikakis, Theodore Dalamagas, and Timos Sellis, "GoNTogle: a tool for semantic annotation and search." The Semantic Web: Research and Applications. Springer Berlin Heidelberg, 2010. 376-380.

[10] Bikakis, Nikos, Giorgos Giannopoulos, Theodore Dalamagas, and Timos Sellis, "Integrating keywords and semantics on document annotation and search." In On the Move to Meaningful Internet Systems, OTM 2010, pp. 921-938. Springer Berlin Heidelberg, 2010.

[11] Popov, Borislav, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyano, and Miroslav Goranov, "KIM semantic annotation platform." In The Semantic Web-ISWC 2003, pp. 834-849. Springer Berlin Heidelberg, 2003.

[12] Handschuh, Siegfried, Steven Staab, and Fabio Ciravegna, "S-CREAM semi-automatic creation of metadata." In Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, pp. 358- 372. Springer Berlin Heidelberg, 2002.

[13] Dill, Stephen, Nadav Eiron and David Gibson, "SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation." In Proceedings of the 12th international conference on World Wide Web, pp.178-186. ACM, 2003.

[14] Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni, "Open information extraction for the web." In IJCAI, vol. 7, pp. 2670-2676. 2007.

[15] Kogut, Paul A., and William S. Holmes III, "AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages." Semannot@ K-CAP, 2001.

[16] Laclavik, Michal, "Ontology based text annotation-OnTeA." Frontiers in Articial Intelligence and Applications 154 (2007): 311.

[17] Pudota, Nirmala, "Automatic keyphrase extraction and ontology mining for contentbased tag recommendation." International Journal of Intelligent Systems 25.12, 2010

[18] Khattak, Asad Masood, N. Ahmad, Jibran Mustafa, "Context-Aware Search in Dynamic Repositories of Digital Documents." Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on. IEEE, 2013.

[19] Lu, Yiyao, "Annotating search results from Web databases." Knowledge and Data Engineering, IEEE Transactions on 25.3 (2013): 514-527.

[20] Harith, "Automatic ontology-based knowledge extraction and tailored biography generation from the web." IEEE Intelligent Systems 18.1 (2003): 14-21.

[21] Iwata, Tomoharu, Takeshi Yamada, and Naonori Ueda, "Modeling Noisy Annotated Data with Application to Social Annotation." Knowledge and Data Engineering, IEEE Transactions on 25.7 (2013): 1601-1613.

[22] Brut, Mihaela M., Florence Sedes, and Stefan Daniel Dumitrescu, "A semantic-oriented approach for organizing and developing annotation for e-learning." Learning Technologies, IEEE Transactions on 4.3 (2011): 239 - 248.

[23] Handschuh, Siegfried, Raphael Volz, and Steven Staab, "Annotation for the deep Web." IEEE Intelligent Systems 18.5 (2003): 42-48.

[24] Heflin, Jeff, and James Hendler, "A portrait of the Semantic Web in action." Intelligent Systems, IEEE 16.2 (2001): 54-59.

[25] Vargas-Vera, Maria, "MnM: Ontology driven semi-automatic and automatic support for semantic markup." Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. Springer Berlin Heidelberg, 2002. 379-391.

[26] Vagelis Hristidis and Eduardo J. Ruiz, "Cads: A collaborative adaptive data sharing platform", VLDB Workshop on Personalized Access, Profile Management, and Context Awareness in Databases (PersDB 2009).

[27] Ruiz, Eduardo J., Vagelis Hristidis, and Panagiotis G. Ipeirotis. "Facilitating Document Annotation using Content and Querying Value." Knowledge and Data Engineering, IEEE Transactions on 26.2 (2014): 336-349.