Systematic Review of Text Mining Application Using Apache UIMA

Ida Bagus Gede Purwania[1], I Nyoman Satya Kumara[2], and Made Sudarma[3]

[1,2,3] Department of Electrical and Computer Engineering, Post Graduate Program, Udayana University
*guspurwania@gmail.com

Abstract Companies are often faced with a number of data and information in the form of unstructured texts. The unstructured data set can be processed / extracted so that it can benefit the company in the decision making process or strategy that must be carried out by the company. Text Mining is one solution to overcome these problems. Text Mining can be defined as the process of retrieving information sourced from several documents. One of the most commonly used text mining tools is Apache UIMA. This study aims to systematically study literature on the implementation of text mining and Apache UIMA by using several related databases, including reviewing text mining, Apache UIMA, and reviewing journal of text mining and Apache UIMA. These journals are reduced using certain criteria. The results obtained are the 20 journals that discuss the implementation of text mining and Apache UIMA. Based on the analysis of these journals, it can be concluded that the application of Text Mining is more widely used in the field of Classification with the method often used is Naive Bayes Classifiers. The average accuracy of the method reaches more than 85%, which means the method is very effective for classification. Specifically, Apache UIMA is more widely implemented in the Information Extraction and NLP fields. The main component of Apache UIMA that is often used is the Annotator Engine and is very effectively implemented for information extraction.

***Index Terms*—Systematic Review, Unstructured Text, Text Mining, Apache UIMA, Implementation Review.**

I. Introduction

The information age in which we now live is characterized by the rapid growth of data and information that is collected, stored and made available on electronic media. Companies are often faced with a number of data and information in the form of unstructured texts, such as complaints data or customer satisfaction data. According to studies by Merrill Lynch and Gartner, 85 to 90 percent of all company data is taken and stored in the form of unstructured data [1]. The unstructured data set can be processed / extracted so that it can benefit the company in the decision making process or strategy that must be carried out by the company. The process of extracting data from text data to get insight certainly cannot be done conventionally because it will require enormous effort.

Text Mining is one solution to overcome these problems. Text Mining can be defined as the process of retrieving information sourced from several documents. Text mining is a technique for analyzing large amounts of natural language text, and detecting lexical patterns to extract useful information, text mining is used to find interesting information from very large databases [2]. Conceptually text mining is a technique used to deal with classification, clustering, information extraction and information retrieval problems [3]. Text mining is widely used to find hidden patterns and information in a large number of semi and unstructured texts [4]. Text mining or knowledge discovery is a subprocess of data mining, which is widely used to find hidden patterns and information in a large number of unstructured texts [4]. Text mining extracts interesting patterns to explore knowledge from textual data sources, the source of this research is sourced from the web which produces a lot of textual content with diverse structures [3].

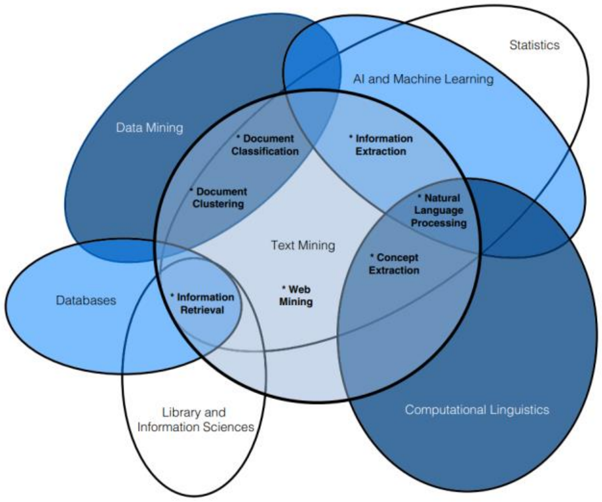

Fig. 1. Diagram of the intersection of text mining and six related fields

Text mining is a multi-disciplinary approach or approach based on information search, data mining, machine learning, statistics, and linguistic computing [3].

The comprehensive definition of Text Mining is not clearly explained, because the field arises from a group of related but different disciplines. Figure 1 shows the other six main areas that intersect with text mining, because of the breadth and disparity of the contributing scientific disciplines it is difficult for experts to characterize concisely [5]. The six areas include database, data mining, statistics, AI & Machine Learning, Computational Linguistics, and Library & Information Sciences. In addition there are seven areas of text mining practice that slice each of the related fields namely Information Retrieval, Web Mining, Concept Extraction, Natural Language Processing, Information Extraction, Document Classification, and Document Clustering [5].

InfoSphere Warehouse provides text analysis functions which are based on Unstructured Information Management Architecture (UIMA), namely Apache UIMA. Unstructured information represents the largest, most recent and fastest-growing source of information available for business and government. With Apache UIMA, companies can analyze large volumes of unstructured information to find, organize, and provide relevant knowledge to decision makers. Unstructured data must be analyzed to interpret, detect, and find interesting concepts that are not explicitly marked or explained in the original document. By analyzing unstructured content, UIMA applications utilize various analysis technologies including Natural Language Processing (NLP), Information Retrieval (IR), Machine Learning, Ontologies, Automated Reasoning, and Knowledge Resource. This technology was developed independently by special scientists and engineers who use different techniques, interfaces and platforms [6].

II. Literature Review

A. Text Mining

Text mining and text analysis are general terms that describe various technologies for analyzing and processing semi-structured and unstructured text data. The equation behind each of these technologies is the need to "convert text into numbers" so that a powerful algorithm can be applied to a large document database. Text mining is one of the special fields in data mining which has the definition of mining data in the form of text where data sources are usually obtained from documents and the purpose is to search for words that can represent the contents of documents [7].

Text mining can analyze documents, group documents based on the words contained in them, and determine the similarity between documents to find out how they relate to other variables [8]. The most common applications of text mining today are spam filtering, sentiment analysis, measuring customer preferences, summarizing documents, grouping research topics, and many others.

Mining the text can be divided into seven areas of practice, based on the unique characteristics of each field as shown in Figure 2. Although different, these areas are highly interrelated. Typical text mining projects will require techniques from various fields. The seven areas of practice include the following [5].

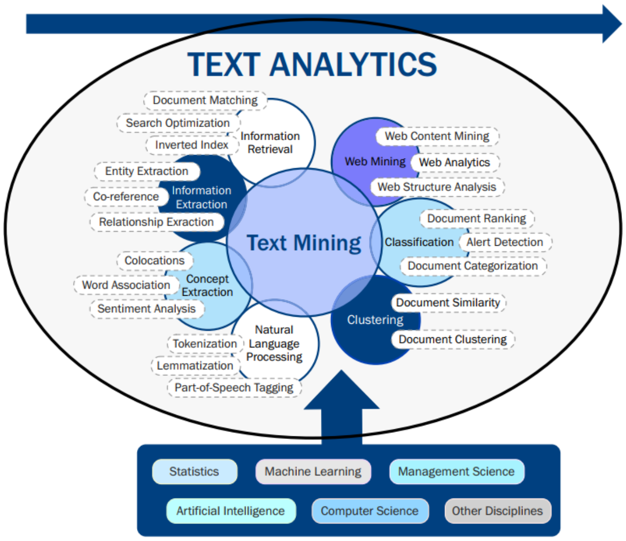

Fig. 2. Visualizing the seven text mining practice areas

1) Information Retrieval (IR) is a method used to retrieve information that is relevant to the user needs of a collection of information automatically [9]. The IR process generally deals with finding information whose contents are not structured. Likewise, a user's search keyword called query is also a form that is not structured. This is what distinguishes IR from database systems. Documents are examples of unstructured information. The contents of a document are generally in the form of a collection of texts that are very dependent on the author of the document [10].
2) Document Clustering is an activity of grouping documents based on the characteristics contained therein. The document clustering analysis process basically has two stages: the first transforms documents into quantitative data and the second analyzes documents in the form of quantitative data with the specified clustering method. For the second stage of the process there are various types of clustering methods that can be used. Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. Examples of document clustering include web document clustering for search users [11].
3) Document Classification is a service used to analyze documents and propose classification of documents based on the classification model specified. The services include training and inference capabilities to fit using the dataset model. Document Classification helps implement machine learning to automate the management and processing of large numbers of

business documents with a special classification model. Documents that can be analyzed such as company mailboxes, contract management, invoices, and others.

4) Web mining is the extraction of important and useful patterns but is implicitly stored in a relatively large data set on the world wide web service. Web mining consists of three parts, namely: web content mining, web structure mining, and web usage mining [12]. Web mining technique was first introduced by Etzioni Oren in 1996. Web mining can be defined as an attempt to implement data mining techniques to explore and then study or extract. useful information
5) Information Extraction is a system for finding specific data in natural language text. Information extraction is done by changing the unstructured text into information in a structured form. Data to be extracted is usually obtained from a template in the form of a form or table that will be filled with sentences or components of the text.
6) Natural language processing (NLP) is a computerized approach to analyzing texts based on aspects of theory and technology. NLP is defined as a theoretical field of computational techniques used to analyze and represent naturally written text (human language) at one or more levels of linguistic analysis with the aim of obtaining human-like language processing that can be implemented in various fields [13].
7) Concept Extraction is the technique of mining the most important topic of a document. Concept mining is an activity that results in the extraction of concepts from artifacts. Solutions to the task typically involve aspects of artificial intelligence and statistics, such as data mining and text mining. Because artifacts are typically a loosely structured sequence of words and other symbols (rather than concepts), the problem is nontrivial, but it can provide powerful insights into the meaning, provenance and similarity of documents.

The benefits of text mining are felt in fields that have a lot of text data, such as law (court orders), academic research (scientific articles), finance (quarterly reports), medicine / medicine, biology (molecular interactions), technology (patent files), and marketing (customer comments). For example, various types of interactions with text-based customers in a haphazard format of complaints (or perhaps praise) and guarantee claims can be used to objectively identify product and service characteristics that are considered imperfect to be used as input to product development and allocation service. Likewise, with a variety of programs to reach markets that produce large amounts of data. By not limiting feedback on products and services in a formatted form, customers can present in their own words what they think about the company's products and services. Another area where automatic processing of unstructured text has brought various impacts is in e-mail and electronic communication. Text mining can not only be used to classify and filter junk emails, but can also be used to prioritize emails automatically based on their importance and also generate automatic responses [14].

To be successful, various studies of text mining should follow a good methodology based on 'best practices'. A standard process model is needed that is similar to CRISP-DM, which is the industry standard for data mining projects. Although most CRISP-DM can also be applied to mining text projects, certain process models for text mining will include a variety of data processing activities that are far more complicated.

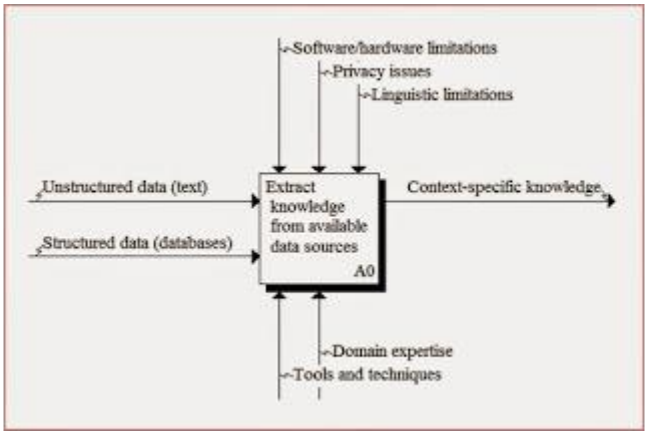
Software/hardware limitations
Privacy issues
Linguistic limitations
Unstructured data (text)
Structured data (databases)
Extract knowledge from available data sources
A0
Context-specific knowledge
Domain expertise
Tools and techniques

Fig. 3. Text Mining Context Diagram

Figure 3 illustrates the high-level context diagram of a text mining process in general. This context diagram presents the scope of the process, emphasizing some of its interfaces with the larger environment. Basically, the picture explains the boundaries around certain processes to identify explicitly what will be included (and excluded) from the text mining process.

As the context diagram shows, the input part (inward arrow to the left of the box) in the process of finding text-based 'knowledge' is 'unstructured' and 'structured' data that is collected, stored and made available for the process. The output part (the arrow to the right of the box) of the process is knowledge with a specific context that can be used for the decision making process. Various kinds of controls (controls) or also called constraints (arrow into the top of the box), from the above process include various software and hardware restrictions, issues about privacy, and various difficulties related to text processing presented in the form of language natural. The mechanism (inner arrow at the bottom of the box) of the above process includes a variety of appropriate techniques, various software tools, and domain expertise. The main purpose of text mining (in the context of knowledge discovery) is to process data (text) that is not structured (and also structured data, if any and relevant to the problem being highlighted) to extract various patterns that can be followed up and meaningful for the process of taking better decision.

B. Apache UIMA

Unstructured information represents the largest, most recent and fastest-growing source of information available in the business and government environment. High-value content in this large collection of unstructured information cannot be utilized. Finding what is needed or mining data on unstructured information sources presents new challenges.

The Unstructured Information Management Architecture (UIMA) is an architecture and software framework for creating, finding, compiling, and using various multi-modal analysis capabilities and integrating them with search technology [15]. Unstructured Information Management Application (UIM) is a software system that analyzes unstructured information such as text, audio, video, images, etc., to find, organize, and provide relevant knowledge to users [16].

First and foremost, the unstructured data must be analyzed to interpret, detect and locate concepts of interest, for example, named entities like persons, organizations, locations, facilities, products etc., that are not explicitly tagged or annotated in the original artifact. More challenging analytics may detect things like opinions, complaints, threats or facts. And then there are relations, for example, located in, finances, supports, purchases, repairs etc. The list of concepts important for applications to discover in unstructured content is large, varied and often domain specific. Many different component analytics may solve different parts of the overall analysis task. These component analytics must interoperate and must be easily combined to facilitate the developed of UIM applications.

The result of analysis are used to populate structured forms so that conventional data processing and search technologies like search engines, database engines or OLAP (On-Line Analytical Processing, or Data Mining) engines can efficiently deliver the newly discovered content in response to the client requests or queries [15].

In analyzing unstructured information, UIM applications utilize a variety of analysis technologies, including statistical and rule-based Natural Language Processing (NLP), Information Retrieval (IR), machine learning, and ontology. The UIMA framework provides a run-time environment where developers can connect and run their UIMA component implementation, together with components that are developed independently, and with which they can build and use UIM applications. This framework is not specific to any IDE or platform.

UIMA as a bridge from the unstructured world to the structured world and supports the creation, discovery, compilation and dissemination of various analytical capabilities and connecting them with structured information services as shown in Figure 4 [15].

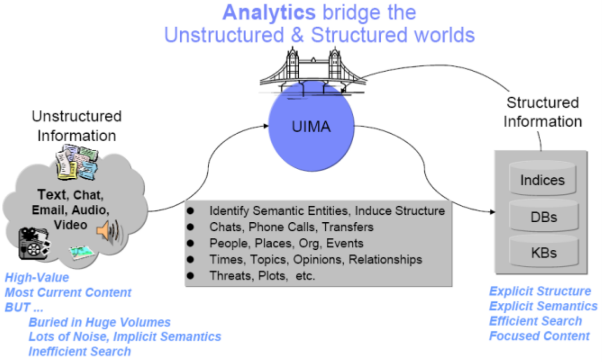

Fig. 4. UIMA as a analytics bridge from the unstructured world to the structured world

UIMA allows the development team to match the right skills with the right part of the solution and helps enable rapid integration across technologies and platforms using a variety of different placement options. These range from tightly paired deployments for high performance, single machines, embedded solutions to parallel and fully distributed deployments for highly flexible and measurable solutions.

UIMA allows applications to be broken down into components, for example "language identification" => "language-specific segmentation" => "sentence boundary detection" => "entity detection (person / place name, etc.)". Each component implements an interface that is determined by the framework and provides metadata that explains itself through an XML descriptor file. The framework manages these components and the data flow between them. Components written in Java or C ++; data flowing between components is designed for efficient mapping between these languages [17]. UIMA also provides the ability to wrap components as network services, and can scale very large volumes by replicating processing pipes through a group of network nodes.

Apache UIMA is an open source platform under the Apache license, the implementation of the UIMA framework that provides a common platform for industry and academia to collaborate in finding important knowledge from information sources in accelerating technological development worldwide. This technology, UIMA SDK (Software Development Tool), is a JavaTM implementation of the UIMA framework, and supports the implementation, description, composition, and deployment of UIMA components and applications. It also supports developers with an Eclipse-based development environment that includes a set of tools and utilities for using UIMA.

UIMA is an architecture in which basic building blocks called Analysis Engines (AEs) are composed to analyze a document and infer and record descriptive attributes about the document as a whole, and/or about regions therein. This descriptive information, produced by AEs is referred to generally as analysis results. Analysis results typically represent meta-data about the document content. One way to think about AEs is as software agents that automatically

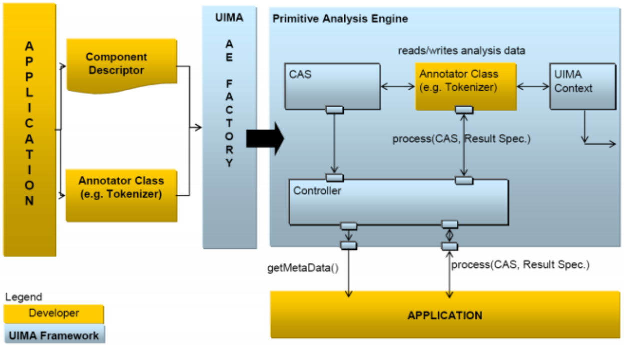
APPLICATION
Component Descriptor
Annotator Class (e.g. Tokenizer)
UIMA
AE FACTORY
Primitive Analysis Engine
reads/writes analysis data
CAS
Annotator Class (e.g. Tokenizer)
UIMA Context
process(CAS, Result Spec.)
Controller
getMetaData()
process(CAS, Result Spec.)
APPLICATION
Legend
Developer
UIMA Framework

Fig. 5. UIMA Framework Analysis Engine (AE)

discover and record meta-data about original content as show in Figure 5 [15].

The UIMA framework treats Analysis engines as pluggable, composible, discoverable, managed objects. At the heart of AEs are the analysis algorithms that do all the work to analyze documents and record analysis results. UIMA provides a basic component type intended to house the core analysis algorithms running inside AEs. Instances of this component are called Annotators. The analysis algorithm developer's primary concern therefore is the development of annotators. The UIMA framework provides the necessary methods for taking annotators and creating analysis engines. How Annotators represent and share their results is an important part of the UIMA architecture. To enable composition and reuse, UIMA defines a Common Analysis Structure (CAS) precisely for these purposes. The CAS is an object-based container that manages and stores typed objects having properties and values. Object types may be related to each other in a single-inheritance hierarchy. Annotators are given a CAS having the subject of analysis (the document), in addition to any previously created objects (from annotators earlier in the pipeline), and they add their own objects to the CAS as show in Figure 6 [15].

The CAS serves as a common data object, shared among the annotators that are assembled for an application. Many UIM applications analyze entire collections of documents. UIMA supports this analysis through its Collection Processing Architecture. This part of the architecture allows specification of a "source-to-sink" flow from a collection reader though a set of analysis engines and then to a set of CAS Consumers. The collection reader's job is to connect to and iterate through a source collection, acquiring documents and initializing CASes for analysis. After the analysis engines have added their information to the CAS, CAS consumers do the final CAS processing, for example, sending the CAS contents to a search engine or extracting elements of interest and populating a relational database. A Semantic Search engine is included in the UIMA SDK; it will allow the developer to experiment with indexing analysis results, which will enable semantic searches using the the annotations in the CAS [15].

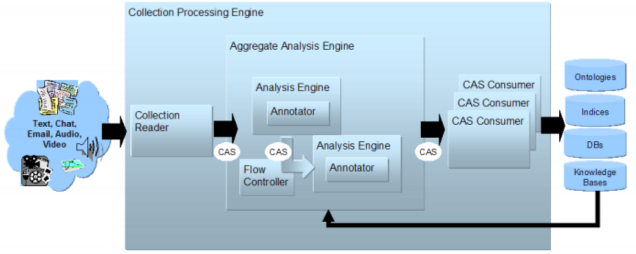

Fig. 6. Collection Processing Engine Architecture

III. Method

The research method used is systematic literature study [18]. This literature study began by collecting several related databases including text mining, text mining technology development, Apache UIMA as a text mining tool, and the implementation of Apache UIMA in text mining applications. The schematic of the study is shown in Figure 7.

The search is performed on credible data sources and journals. The journals collected are first reduced using certain criteria. The criteria used are the completeness of the article. Selected scientific articles are articles published in English and Indonesian in full. Manuscripts are considered intact, if they contain the title, name of the author, publisher, abstract, and there is complete article content to the bibliography. The next criterion is the relevance of the research topic, which is to focus on the implementation of text mining and Apache UIMA.

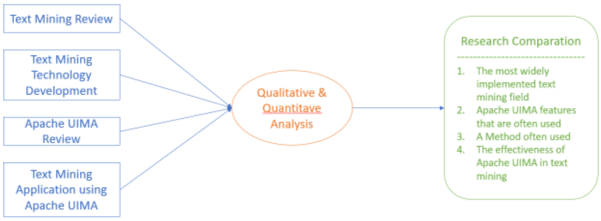
Text Mining Review
Text Mining Technology Development
Apache UIMA Review
Text Mining Application using Apache UIMA
Qualitative & Quantitave Analysis
Research Comparation
1. The most widely implemented text mining field
2. Apache UIMA features that are often used
3. A Method often used
4. The effectiveness of Apache UIMA in text mining

Fig. 7. Schematic Research

The data collected is analyzed using qualitative and quantitative techniques. The expected results of the analysis of this literature study are to be able to compare each scientific article, make quantitative analysis related to the fields, components / features, the method most widely implemented in text mining applications, and the effective use of Apache UIMA in text mining.

IV. Discussion Result

Research related to the implementation of text mining applications has been carried out by academics/ practitioners who have been tested in various fields, with different features and methods. This research will discuss some relevant research related to the application of text mining using Apache UIMA. Before entering the Apache UIMA implementation, there are several studies that can be used as a reference in developing text mining applications.

Ratniasih et al conducted research on the application of text mining in spam filtering for chat applications [19]. The purpose of this research is to minimize spamming actions in sending messages via chat using text mining and challenge-response filtering techniques. The filtering process is done by text pre-processing and analyzing stages so that the sentences that are stated as spam sentences are obtained. Based on the results of testing 524 message sentences obtained a system accuracy of 91.41%. The application of text mining with NLP in the field of text-based communication is very effectively applied to deal with the problem of spamming, to make improvements to the accuracy of the system more training needs to be done.

Yudiarta et al conducted research on the application of clustering text mining methods to group news on unstructured textual data with the K-Means algorithm [20]. To process the documents obtained in order to make it easier in the clustering process, the document preprocessing process is carried out first through the stages of case folding, tokenization, filtering and stemming. The results of trials conducted using different amounts of data that are 50, 100, 200, 300, 400, and 500 data obtained the results that the K-Means algorithm applied to cluster news is able to work and provide satisfactory accuracy, with an average of average precision of 70.76% while recall of 70.86% and purity of 0.76 for all test data.

Dwi Ardiada et al apply text mining on social media to detect user emotions using the Support Vector Machine and K-Nearest Neighbor methods [21]. To detect the emotion of the text on Twitter social media services with unstructured data it is necessary to do a text analysis using Text Mining. The SVM method is used to classify datasets in determining emotional classes. After obtaining the emotional class label, the dataset is selected again based on the emotional class specified in the emotional class table. The K-NN method will reclassify based on a predetermined dataset. K-NN works by calculating euclidean distance to determine the emotional label that has the closest distance to the test data. Test data conducted by the two methods can produce an average precision value of 0.4564, a recall value of 0.502 and an accuracy value of 0.8104.

Azman Maricar et al apply text mining on Twitter social media to classify racism using a combination of POS Tagging, Naive Bayes Classifier, and K-Nearest Neighbor methods [22]. This study aims to produce computer applications that are able to classify texts into various classes, namely opinions / not opinions, positive / negative opinions, and racism. The dataset used comes from the Twitter application which consists of 600 sentences. Based on the trial results, the application accuracy is 71% for the POS Tagging method, 27% for the Naive Bayes Classifier method, and 41% for the K-Nearest Neighbor method. A potential solution to improve accuracy is to use linguists to prepare datasets that use word level approaches and optimization methods for each method chosen.

Agus Hermanto implements text mining to determine the final assignment categories of students based on their abstracts using the Naive Bayes method [23]. This study aims to assist the final project coordinator in grouping the final project proposal. Naive Bayes method which will be implemented into the final project proposal information system can provide a new solution to determine the final project proposal category based on the abstraction made by students. In the trial results of this method, it can be concluded that it is quite successful and can be broadly used as a tool in classifying final project documents. The accuracy based on testing in the hardware and networking category reached 86%, the information system category reached 80% accuracy and the accounting information system category reached 89%. Overall, based on the number of datasets tested and the level of success achieved, this system has an accuracy rate of 87%.

Taufik Kurniawan implements text mining for twitter user sentiment analysis using the Naive Bayes Classifier method and Support Vector Machine [24]. Sentiment analysis was conducted to find out public sentiments towards television media, namely TV One, Metro TV, and Kompas TV, whether the majority of the public considered positive or negative. Public responses to the mainstream media were obtained from the Application Programming Interface (API) on Twitter. In this study, the text preprocessing used is case folding, tokenizing, stopwords, and stemming. For the stemming preprocessing algorithm, the confix-stripping stemmer algorithm is used. While in the analysis of the classification of the text data the Naïve Bayes Classifier and Support Vector Machine methods are used. Classification using NBC on TV One and Kompas TV media data resulted in an accuracy of 95.6% and 97.8%, while in Metro TV media produced a G-mean and AUC values of 81.3% and 82.36%, respectively. Classification using SVM on TV One and Kompas TV media data produces an accuracy of 97.9% and 99.3%, while in Metro TV media produces G-mean and AUC values respectively of 97.35% and 97.38%.

Dea Herwinda Kalokasari et al implemented text mining using the multinomial naive bayes classifier algorithm in the outgoing mail classification system by taking a case study at DISKOMINFO, Tangerang Regency [25]. The researcher examines the Multinomial Naive Bayes Classifier to classify outgoing mail so that it can determine the letter number automatically. The classification system is supported by confix-stripping stemmer to find basic words and TF-IDF for weighting words. Testing is measured using a confusion matrix. From the test results show that the implementation of the Multinomial Naive Bayes Classifier on the letter classification system has a level of accuracy, precision, recall, and F-measure respectively 89.58%, 79.17%, 78.72%, and 77.05%.

Imam Fahrur Rozi implements text mining for the analysis of public opinion sentiments at universities [26]. Mining is a research branch of text mining with a focus on analyzing the opinion of a text document. In this study an opinion mining system was developed to analyze public opinion in tertiary institutions. In the subjectivity and target detection subprocesses, Part-of-Speech (POS) Tagging is used using Hidden Makov Model (HMM). In the POS Tagging process results are then applied rules to find out whether a document is included as an opinion or not, as well as to find out which part of the sentence is an object that is the target of opinion. Documents that are recognized as opinions are then classified into negative and positive opinions (subprocess opinion orientation) using the Naïve Bayes Classifier (NBC). The test results obtained precission and recall values for the subjectivity document subprocesses are 0.99 and 0.88, for the target detection subprocesses are 0.92 and 0.93, and for opinion orientation subprocesses are 0.95 and 0.94.

Trya Sovi Kartikasari et al implemented text mining for the analysis of public opinion of the presidential candidates. In this study an opinion will be analyzed regarding the electability of the presidential candidates from social media Twitter from Twitter social media using the Naïve Bayes Classifier (NBC) method and determine the factors formed from opinions using Principal Component Analysis (PCA) [27]. Opinion data from social media Twitter was obtained using the keywords "Jokowi" and "Prabowo". Some of these opinions were chosen as training data to obtain negative and positive class sentiments. After the training process, a process is carried out on test data and validation data. Accuracy results for Jokowi topic test data on positive sentiment tweets get an accuracy of 88.63% and a negative of 91.06%. While for Prabowo, the positive sentiment was 88.58% and 80.37% was negative. The average accuracy for the whole topic is 86.89%. To get the factors on each sentiment, the PCA value calculation process is carried out. Each sentiment is then carried out by a factor analysis by experts, which found 20 factors that have been successfully interpreted by experts.

Fitri Handayani and Feddy Setio Pribadi implemented text mining with the native bayes classifier algorithm in the classification of automatic text complaints and public reporting through the call center service 110 of the Indonesian National Police [28]. The Naive Bayes Classifiers method is a method of text classification based on the probability of keywords comparing training documents and test documents. Both are compared through several stages of the equation, which finally obtained the highest probability results set as a new document category. The results of research conducted by researchers, namely the classification of automatic text reporting and public complaints using the Naive Bayes Classifiers method produces a high average accuracy, namely recall 93%, 90% precission, and 92% f-measure.

Based on 10 journals related to the implementation of text mining that have been discussed and analyzed, the comparative results obtained from each journal are shown in Table 1. The results of the analysis in Table 1 show that the application of text mining in research is more utilized in the field of Classification and the method often used is Naive Bayes Classifiers. The average accuracy rate of the research analyzed is above 85% which shows that the method has a very high accuracy for the classification process. Next is to analyze several journals related to the implementation of Apache UIMA in text mining.

TABLE I
TEXT MINING RESEARCH COMPARATION

No	Author	Field	Method	Accuracy
1	Ratniasih et al (2017)	NLP	Text Pre-Processing	91,41%
2	Yudiarta et al (2018)	Clustering	K-Means	>70%
3	Dwi Ardiada et al (2019)	NLP & Classification	SVM & K-NN	81%
4	Azman Maricar et al (2019)	Classification	POS Tagging, NBC, K-NN	27 – 71%
5	Agus Hermanto (2016)	Classification	NBC	87%
6	Taufik Kurniawan (2017)	NLP & Classification	NBC & SVM	81 – 99%
7	Dea Herwinda et al (2018)	Classification	NBC	89,58%
8	Imam Fahrur Rozi (2012)	NLP & Classification	POS Tagging & NBC	88 – 99%
9	Trya Sovi et al (2018)	NLP & Classification	NBC & PCA	86,89%
10	Fitri Handayani et al (2015)	Classification	NBC	> 90%

Putu Wuri Handayani et al applied Apache UIMA in their research to develop a semantic based search engine for Indonesian [29]. The purpose of this research is to develop a search engine that can analyze Indonesian text content with Natural Language Processing and Semantic Web. Apache UIMA is implemented in analyzing text content. The semantic data definition for each article was made using XML integrated with UIMA. UIMA has several main components to carry out text content analysis using predefined semantic data such as Collection Reader, Analysis Engine, and CAS Consumer. Collection Reader functions to collect all text files to be analyzed and returns CAS type which includes articles to be analyzed. Then, the Analysis Engine uses the CAS to analyze text content and produce CAS tags that are rich in tags. Furthermore, CAS Consumer uses the CAS to generate several tags for each article. The tags generated for each article will be stored in a database by prototype to speed up the search process. The result is the use of semantic analysis makes it easy for users to find the articles they need.

Renaud Richardet et al implemented UIMA as an NLP Toolkit for Neuroscience called Bluima [30]. Bluima is a Natural Language Processing (NLP) processing pipeline that focuses on the extraction of neuroscientific content based on the UIMA framework. Bluima is built based on models from biomedical NLP (BioNLP) such as tokenizers and special lemmatizers. It adds further models and tools specifically for neuroscience (for example entities known as neurons or brain regions) and provides collection readers for neuroscientific corporations. Two novel UIMA components are proposed: the first allows configuration and installation of UIMA pipelines using a simple scripting language, allowing non-UIMA experts to design and run UIMA pipelines. The second component is the general structure analysis shop (CAS) based on MongoDB, to carry out additional annotations of large corporate documents.

Carlos Rodriguez-Penagos et al implemented UIMA and Solr for sentiment analysis and visualization[31]. Sentiment analysis uses hotel customer review data by extracting opinion objects or polarity attributes using various UIMA modules including UIMA Collection Tools, OpenNLP, Lemmatizer, JNET, DeSR, DependencyTreeWalker, and Weka Wrapper. Solr-based graphical interface is used to explore and visualize the collection of reviews and opinions expressed therein. The results of tests conducted with 700 OU manually annotated by 3 independent reviewers resulted in the truth value of OU identified by the system 88.5%, while the average polarity level was 70%. The combination of UIMA and Solr allows for the development of a very flexible platform that makes it easy to integrate and combine processing modules from various sources and in various programming languages, as well as navigate and visualize results easily and efficiently.

Jan Stadermann et al implemented Apache UIMA to extract hierarchical data points and tables from scanned semi-structured contracts[32]. It consists of a collection of small and simple Analysis Components that extract increasingly complex information based on previous extractions. This technique is applied to extract each data point and table. Each type of expert is implemented as a configurable annotation engine. The whole extraction system consists of a large hierarchy of analysis machines, which includes several hundred elements. Type systems, in contrast, only consist of three main types, namely for simple fields, tables, and table rows. Annotation types, extracted values, etc. are stored as features. Final and intermediate explanations are represented by these types. The experiment showed 97% overall precision with a 93% recall about simple data points and 89% / 81% for table cells measured against the basic truth that was entered manually. Because of its modular nature, this system can easily be expanded and adapted to other contract collections as long as several data models can be formulated.

Yoshinobu Kano et al developed an integrated language resource evaluation platform with a comprehensive UIMA resource library called U-Compare [33]. These resources can be used as local services or web services, and may even be hosted on a cluster machine to improve performance, while users do not need to be aware of these differences. In addition to the resource library, an integrated language processing platform is provided, which allows the creation of workflows, comparisons, evaluations and visualizations, using resources in any library or UIMA component, without any programming through a graphical user interface, while the command line launcher is also available without a GUI. The evaluation itself is processed in the UIMA component, users can create and connect their own evaluation metrics in addition to predetermined metrics. U-Compare has been used successfully in many projects including BioCreative, Conll and joint assignments with BioNLP.

Min Jiang et al conducted a study to extract and standardize drug information in clinical texts using a UIMA named MedEx-UIMA System[34]. MedEx was built using the Java language with the UIMA framework as a pipeline based system, which defined several classes including Sentence Boundary Detector, Tokenizer, Section Tagger, Semantic Tagger, Parser and Encoder. In this system a new encoding module is developed that maps the names of drugs/doses/forms extracted for general and specific RxNorm concepts and translates drug frequency information to ISO standards. This study tested two versions of the MedEx system, the UIMA version and the Python version, by processing 826 documents. Using two manually annotated test sets containing 300 drug entries from the drug list and 300 drug entries from the narrative report, the MedEx-UIMA system achieves an F-measure of 98.5% and 97.5% to encode the drug name into the drug ingredient the appropriate generic RxNorm, and Size-F respectively 85.4% and 88.1% to map the name/dose/form of the drug with the most specific RxNorm concept. It also reaches an F-size of 90.4% to normalize frequency information to ISO standards.

Matthias Grabmair et al conducted experiments in conceptual legal decision making using the UIMA System and Tools [35]. The system developed is named LUIMA. This research presents the first results of the feasibility trial evidence in the taking of conceptual legal documents in a particular domain. Conceptual document markup is done automatically using LUIMA, a special semantic legal extraction tool box based on the UIMA framework. The system consists of modules for automatic sub-sentence level annotations, machine learning based sentence annotations, basic retrieval using Apache Lucene and re-learning based on machine rankings of documents taken. In a one-time leave trial on a limited corpus, the resulting rank scores higher for most of the queries tested than the base ranking created using the full legal information system of commercial text.

Steven Bethard et al developed the ClearTK 2.0 application, which is a design pattern for Machine Learning at UIMA[36]. ClearTK adds machine learning functionality to the UIMA framework, provides wrappers for popular machine learning libraries, feature-rich extraction libraries that work in a variety of different classifiers, and utilities for implementing and evaluating machine learning models. Since its founding in 2008, ClearTK has grown in response to feedback from developers and the community. This evolution has followed a number of important design principles including: a conceptually simple annotator interface, legible pipelines description, minimal collection reader, type system agnostic code, modules that are set up for ease of import, and help users understand the complex UIMA framework.

Maciej Ogrodniczuk implements UIMA for Polish language processing[37]. The chain tool generates multi-level UIMA encoded text annotations that can be used by high-level Web applications for complex language-intensive operations such as automatic categorization, information extraction, machine translation or summaries. This research focuses on the characteristics of integrated processing chain languages for Polish and specific findings for this integration. The inflection characteristics of Poland offer the possibility to present some more advanced functions such as the multi-word lematization, important for the real-life presentation of extracted phrases.

Jennifer H. Garvin et al. Conducted a study for automatic ejection fraction (EF) extraction for quality measurements using regular expressions on UIMA for heart failure [38]. The purpose of this study is to build a natural language processing system to extract EF from echocardiogram reports to automate reporting measurements and to validate system accuracy using comparative reference standards developed through human review. In that study a series of regular expressions and rules were made to capture EF using a 765 random sample of echocardiograms from seven VA medical centers. The documents were randomly assigned to two sets: one set of 275 was used for training and the second set of 490 was used for testing and validation. To set a reference standard, two independent reviewers annotated all documents in both sets. The system test results for the classification of EF document level <40% have a sensitivity (recall) of 98.41%, specificity of 100%, positive predictive value (precision) of 100%, and an F size of 99.2%. The system test results at the concept level have a sensitivity of 88.9% (95% CI 87.7% to 90.0%), a positive predictive value of 95% (95% CI 94.2% to 95.9%), and a measure of F 91.9% (95% CI 91.2% to 92.7%). An automatic information extraction system can be used to accurately extract EF for quality measurements.

Apache UIMA has been implemented in various fields such as in the fields of language, science, health, enterprise, law, and so on. Based on 10 journals that have been discussed related to the implementation of Apache UIMA in text mining, it can be analyzed and grouped a review of each journal shown in Table 2.

The results from Table 2 show that the application of the UIMA Application is more widely used in the field of Information Extraction, and Natural Language Processing. The most commonly used UIMA component is the Annotation Engine which is used to extract information based on a specified / regulated regular expression. Based on related studies that apply Apache UIMA, it can be concluded that Apache UIMA is very effective for Text Mining, especially for information extraction.

TABLE 2
UIMA IMPLEMENTATION RESEARCH COMPARATION

No	Author	Field	Component	Effective
1	Handayani et al (2015)	NLP	Collection Reader, Analysis Engine, and CAS Consumer	√
2	Richardet et al (2013)	NLP	UIMA Pipelines & CAS	√
3	Rodriguez-Penagoz et al (2013)	IE	UIMA Collection Tools	√
4	Stadermann et al (2013)	IE	Annotation Engine	√
5	Kano et al (2010)	NLP	UIMA Resource Library	√
6	Jiang et al (2014)	IE	UIMA Pipelines	√
7	Grabmair et al (2015)	IE	Annotation Engine	√
8	Bethard et al (2014)	IE & Classification	UIMA Resource Library	√
9	Ogrodniczuk (2011)	NLP	Annotation Engine	√
10	Garvin et al (2012)	IE	Annotation Engine	√

V. CONCLUSION

Text Mining is the most effective way to manage and analyze unstructured documents / information to gain insight in decision making. Text Mining is widely applied by researchers to classify. The method often used is Naive Bayes Classifiers because it has a high accuracy of classification results.

Apache UIMA is one of the tools used in Text Mining. Apache UIMA has been applied in various fields such as science, health, language, enterprise, law, and so on. Apache UIMA is often used for information extraction using the Annotation Engine owned by Apache UIMA. Users can make rules/regular expressions according to their needs in data extraction. Apache UIMA is very effective for text mining and text analysis.

REFERENCES

[1] W. McKnight, “Text Data Mining is Business Intelligence,” *Information Management Magazine*, 2005.

[2] R. Janani and S. Vijayarani, “Text Mining Research: A Survey,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, no. 4, p. 6564, 2016.

[3] R. Talib, M. Kashif, S. Ayesha, and F. Fatima, “Text Mining: Techniques, Applications and Issues,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 414–418, 2016, doi: 10.14569/ijacsa.2016.071153.

[4] A. K. Naithani, “A Comprehensive Study of Text Mining Approach,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 2, p. 69, 2016.

[5] G. Miner, D. Delen, J. Elder, A. Fast, T. Hill, and R. Nisbet, “The Seven Practice Areas of Text Analytics,” in *The Seven Practice Areas of Text Analytics*, no. January, 2012, pp. 29–41.

[6] IBM, “UIMA concepts,” *IBM Knowledge Center*. https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.7.0/com.ibm.datatools.datamining.doc/c_ta_uima_concepts.html (accessed May 19, 2020).

[7] R. J. Mooney, “Machine Learning Text Categorization,” University of Texas, Austin, 2006.

[8] Statsoft, “Text Mining Introductory Overview,” 2015. http://www.statsoft.com/textbook/text-mining (accessed May 20, 2020).

[9] H. Bunyamin, “Algoritma Umum Pencarian Informasi Dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen,” *J. Inform.*, vol. 1, no. 2, 2015.

[10] J. Pardede, “Implementasi Multithreading Untuk Meningkatkan Kinerja Information Retrieval Dengan Metode GVSM,” *J. Sist. Komput.*, vol. 4, no. 1, pp. 1–6, 2014.

[11] G. Salton, *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. New York: Cornell University, Addison Wisley Publishing Comp, 1989.

[12] N. Tyagi, Kumar, Solanki, and M. Wadhwa, “Analysis of Server Log by Web Usage Mining for Website Improvement,” *Int. J. Comput. Sci.*, vol. 7, no. 4, 2010.

[13] E. D. Liddy, *Natural Language Processing*, 2nd Ed. New York: Marcel Decker, Inc, 2001.

[14] S. . Weng and C. K. Liu, “Using Text Classification and Multiple Concepts to Answer Emails,” *Expert Syst. with Appl.*, vol. 26, no. 4, pp. 529–543, 2004.

[15] Apache UIMA Development Community, *UIMA Overview & SDK Setup*, Version 3. The Apache Software Foundation, International Business Machines Corporation, 2019.

[16] IBM, “Unstructured Information Management Architecture SDK,” *IBM Developer*. https://www.ibm.com/developerworks/data/downloads/uima/index.html (accessed May 20, 2020).

[17] The Apache Software Foundation, “Apache UIMA,” 2018. https://uima.apache.org/ (accessed May 20, 2020).

[18] F. Espitia, J. . Sánchez-Torres, and E. Galvis-Lista, “Systematic Literature Review of the Implementation of Knowledge Codification Process,” in *European Conference Knowledge Management*, 2016.

[19] N. L. Ratniasih, M. Sudarma, and N. Gunantara, “Penerapan Text Mining Dalam Spam Filtering Untuk Aplikasi Chat,” *Maj. Ilm. Teknol. Elektro*, vol. 16, no. 3, p. 13, 2017, doi:

10.24843/mite.2017.v16i03p03.

[20] N. G. Yudiarta, M. Sudarma, and W. G. Ariastina, “Penerapan Metode Clustering Text Mining Untuk Pengelompokan Berita Pada Unstructured Textual Data,” *Maj. Ilm. Teknol. Elektro*, vol. 17, no. 3, p. 339, 2018, doi: 10.24843/mite.2018.v17i03.p06.

[21] I. M. D. Ardiada, M. Sudarma, and D. Giriantari, “Text Mining pada Sosial Media untuk Mendeteksi Emosi Pengguna Menggunakan Metode Support Vector Machine dan K-Nearest Neighbour,” *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, p. 55, 2019, doi: 10.24843/mite.2019.v18i01.p08.

[22] M. A. Maricar, N. S. Kumara, and M. Sudarma, “Opinion Mining on Twitter Social Media to Classify Racism Using Combination of POS Tagging , Naive Bayes Classifier , and K-Nearest Neighbor,” *Int. Conf. Smart-Green Technol. Electr. Inf. Syst.*, no. October 2018, pp. 25–27, 2019.

[23] A. Hermanto, “Implementasi Text Mining Menggunakan Naive Bayes Untuk Penentuan Kategori Tugas Akhir Mahasiswa Berdasarkan Abstraksinya,” *Konvergensi*, vol. 11, no. 01, 2016, doi: 10.30996/konv.v12i2.1310.

[24] T. Kurniawan, “Implementasi Text Mining Pada Analisis Sentimen Pengguna Twitter Terhadap Media Mainstream Menggunakan Naïve Bayes Classifier Dan Support Vector Machine Media Mainstream Menggunakan Naïve Machine,” Institut Teknologi Sepuluh Nopember Surabaya, 2017.

[25] A. H. Setianingrum, D. H. Kalokasari, and I. M. Shofi, “Implementasi Algoritma Multinomial Naive Bayes Classifier,” *J. Tek. Inform.*, vol. 10, no. 2, pp. 109–118, 2018, doi: 10.15408/jti.v10i2.6822.

[26] I. F. Rozi, S. H. Pramono, and E. A. Dahlan, “Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi,” *Electr. Power, Electron. Commun. Control. Informatics Semin.*, vol. 6, no. 1, pp. 37–43, 2012.

[27] T. S. Kartikasari *et al.*, “Implementasi Text Mining Untuk Analisis Opini Publik Terhadap Calon Presiden,” *Jurnla Simantec*, vol. 7, no. 1, 2018.

[28] F. Handayani and S. Pribadi, “Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110,” *J. Tek. Elektro*, vol. 7, no. 1, pp. 19–24, 2015.

[29] P. W. Handayani, I. M. Wiryana, and J.-T. Milde, “MESIN PENCARI BERBASISKAN SEMANTIK UNTUK BAHASA INDONESIA,” *J. Sist. Inf. MTI-UI*, vol. 4, pp. 110–114, 2012.

[30] R. Richardet, J. C. Chappelier, and M. Telefont, “Bluima: A UIMA-based NLP toolkit for neuroscience,” *CEUR Workshop Proc.*, vol. 1038, no. May, pp. 34–41, 2013.

[31] C. Rodríguez-Penagos, D. G. Narbona, G. M. Sanabre, J. Grivolla, and J. C. Filbá, “Sentiment Analysis and visualization using UIMA and Solr,” *CEUR Workshop Proc.*, vol. 1038, no. Ml, pp. 42–49, 2013.

[32] J. Stadermann, S. Symons, and I. Thon, “Extracting hierarchical data points and tables from scanned contracts,” *CEUR Workshop Proc.*, vol. 1038, pp. 50–57, 2013.

[33] Y. Kano, R. Dorado, L. McCrohon, S. Ananiadou, and J. Tsujii, “U-Compare: An integrated language resource evaluation platform including a comprehensive UIMA resource library,” *Proc. 7th Int. Conf. Lang. Resour. Eval. Lr. 2010*, pp. 428–434, 2010.

[34] M. Jiang, Y. Wu, A. Shah, P. Priyanka, J. C. Denny, and H. Xu, “Extracting and standardizing medication information in clinical text - the MedEx-UIMA system.,” *AMIA Jt. Summits Transl. Sci. proceedings. AMIA Jt. Summits Transl. Sci.*, vol. 2014, no. RxCUI 20610, pp. 37–42, 2014, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25954575%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4419757.

[35] M. Grabmair *et al.*, “Introducing LUIMA: An experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools,” *Proc. Int. Conf. Artif. Intell. Law*, vol. 08-12-June-2015, pp. 69–78, 2015, doi: 10.1145/2746090.2746096.

[36] S. Bethard, P. Ogren, and L. Becker, “ClearTK 2.0: Design Patterns for Machine Learning in UIMA,” *Lr. Int Conf Lang Resour Eval*, 2014, doi: 10.1016/j.physbeh.2017.03.040.

[37] M. Ogrodniczuk, “UIMA-based Language Processing of Polish,” *Advances*, no. January 2011, pp. 2010–2012, 2011.

[38] J. H. Garvin *et al.*, “Automated extraction of ejection fraction for quality measurement using regular expressions in unstructured information management architecture(uima) for heart failure,” *J. Am. Med. Informatics Assoc.*, vol. 19, no. 5, pp. 859–866, 2012, doi: 10.1136/amiajnl-2011-000535.