Multimedia Information Acquisition and Retrieval Enhancement using Intelligent Search System

((تحسين استقصاء واسترجاع معلومات الوسائط المتعددة باستخدام نظام بحث ذكي))

By

Salameh Ahmad Salameh AL-Etaywi

401220078

Supervisor

Dr. Maamoun Khalid Ahmad

Master Thesis Proposal

Submitted In Partial Fulfillment of the Requirement of the Master Degree in Computer Science

Faculty of Information Technology

Middle East University

Amman, Jordan

January 2015

Authorization

I am Salameh Ahmad AL-Etaywi, authorize **Middle East University (MEU)** to provide copies of my dissertation to the concerned libraries, organizations, and institutes upon request.

Name: Salameh Ahmad Salameh AL-Etaywi

Date: 18 /01/2015 .

Signature: ______________

أنا الطالب سلامة أحمد العطيوي، أفوض جامعة الشرق الأوسط بتزويد نسخاً من رسالتي للمكتبات المعنية، المنظمات, الـهيئات والمؤسسات عند طلبها.

الاسم : سلامة أحمد سلامة العطيوي.

التاريخ: 2015/01/18.

التوقيع: ______________

Thesis Committee Decision

This Thesis " Multimedia Information Acquisition and Retrieval Enhancement using Intelligent Search System" was discussed and certified on 18-01-2015.

Thesis Committee		Signature
Prof. Ahmad AL-Kayed	Chairperson	__________
Dr. Ma'amoun Ahmad	Supervisor & Member	__________
Dr. Jehad AL-Sa'di	Member	__________

Acknowledgments

Prior to acknowledgments, I must glorify Allah the Almighty for His blessings who gave me courage and patience to carry out this work successfully, and giving me this opportunity to become a student once again after years of work. I thank Dr. Maamoun Khalid Ahmad for all his help and support; for giving me guidance, encouragement, confidence, and for being receptive to the ideas i came up with. I also wish to express my deepest gratitude to the members of the committee for spending their precious time on reading my thesis and giving me encouragement and constructive comments. Also i would like to thank all information technology faculty members at middle east university, and my parents for everything they had been done for me during my life and for their constant prayers. Finally i would like to thank my wife for her help, support, sacrifice and great patience all the time. I am so grateful to those who helped me in this research work.

Dedication

(وَ إِذْ تَــأَذَّنَ رَبُّــكُمْ لَــئِنْ شَكَرْتُــمْ لَــأَ زِيــدَنَّــكُمْ)

إبــرا هيم ﴿٧﴾

To my kindhearted Father & Mother

To my wonderful wife ... Ala'a

To my dear loving daughter: Jana

To the Soul of My Son: Waseem

I dedicate this work.

Salameh

Table of Contents

List of Abbreviations

API	Application programming interface
ASR	Automatic Speech Recognition
CAD	Computer-Aided Design
CCText	Closed Caption Text
DBS	Database System
HMM	Hidden Markov Models
ICT	Information and Communication Technologies
IR	Information Retrieval
KBS	Knowledge Base System
KDD	Knowledge Discovery from Data
LOD	Linked audio data
LSCOM	A Light Scale Concept Ontology for Multimedia
LVCSR	Large-Vocabulary Continuous Speech Recognition
NDVR	Near-duplicate video retrieval
NLP	Natural Language Process
OCR	Optical Character Recognition
RDF	Resource Description Framework
SAPI	Speech Application Programming Interface
SBV	Superbase Form Definition File
SRT	SubRip Subtitle File

UCC	User-Created contents
VTT	The Web Video Text Tracks Format
XML	Extensible Markup Language
XLS	Filename Extension (Microsoft Excel spreadsheet file)

List of Figures

List of Tables

Multimedia Information Acquisition and Retrieval Enhancement Using Intelligent Search System

Prepared by Salameh Ahmad Salameh AL-Etaywi

Supervised by Dr.Maamoun Khalid Ahmad

Abstract

In a rapidly changing world; Information and communication technologies are dominating our everyday lives; Education, social communication, entertainment, business, and health care are all affected by rapid technical evolution. The educational aspect is one of the aspects that most fit with new technology that save time, effort and cost. This thesis which called "Multimedia Information Acquisition and Retrieval Enhancement using Intelligent Search System" describes a system capable of searching inside education video lectures using text-based search queries, built-in synonyms dictionary, and voice query too. This system dealing with audio content of video lectures. This thesis aims to help university, colleges, and schools students in the educational process, and improve the level of self-learning by providing the ability to search and retrieval information from content of educational material especially the archived ones. The study depends on several interrelated developed steps before searching process include: data preprocessing and build synonyms dictionary. Statistical experiments were carried out in this thesis on (16) different videos with (111) different test words and the results improved between (2.7% to 5.9%) of the results obtained by the researcher Sleit. Those results can inspire designing smart search engine that can infer the topic of video lectures and decide the main keywords automatically.

تحسين استقصاء واسترجاع معلومات الوسائط المتعددة باستخدام نظام بحث ذكي

إعداد الطالب سلامة أحمد سلامة العطيوي

إشراف الدكتور مأمون خالد أحمد

ملخص

في العالم سريع التغير؛ لطالما لعبت تكنولوجيا المعلومات والاتصالات دورا مؤثراً وقوياً في حياتنا اليومية. الترفيه, قطاع التعليم, القطاع العسكري, التواصل الاجتماعي و الرعاية الصحية جميعها مجالات تتأثر بشكل كبير بالتطور التقني السريع, وبسبب المرونة والتقدم المستمر؛ يعد قطاع التعليم بطبيعته قادر على الاستفادة من التوجهات التقنية الحديثة. هذا يعني أن قطاع التعليم قادر على الاستفادة من التطبيقات التقنية التي توفر المال والجهد وتقديم الحلول العملية لخدمة كل من المعلم والطالب. هذه الأطروحة والتي تسمى " تحسين استقصاء واسترجاع معلومات الوسائط المتعددة باستخدام نظام بحث ذكي" تقدم وتصف نظام بحث الذكي قادر على البحث داخل المحاضرات المرئية المؤرشفة باستخدام الاستعلامات البحثية المستندة إلى نص, قاموس مضمن, و الاستعلام صوتي أيضاً. هذا النظام يتعامل مع محتوى الصوت في المحاضرات المرئية بعد تطبيق عدة إجراءات خاصة لتجهيز هذا المحتوى. تهدف هذه الأطروحة إلى مساعدة طلاب الجامعات, الكليات, والمدارس في العملية التعليمية وتحسين مستوى التعليم الذاتي لديهم, من خلال توفير قدرة البحث والاسترجاع للمعلومات المطلوبة من محتوى تلك المواد التعليمية وخاصة المؤرشفة منها. تعتمد هذه الدراسة على عدة خطوات تحضيرية مترابطة قبل إجراء عملية البحث تشمل: تجهيز البيانات, وبناء قاعدة بيانات للمرادفات. أجريت التجارب الإحصائية في هذه الأطروحة على (16) فيديو مختلف و (111) كلمة فحص مختلفة. تم تحسين النتائج بمقدار يتراوح مابين (2,7% الى 5,9%) عن النتائج التي حصل عليها الباحث سليط. هذه النتائج يمكن أن تلهم لتصميم محرك بحث ذكي قادر على أن يستنتج موضوع المحاضرات المرئية والكلمات الرئيسية تلقائيا.

CHAPTER ONE

Introduction

1.1. Overview

The information technology in the development of contemporary, represent an era characterized by excitement, challenges and real opportunities to change the pattern of human communication quality and quantity, era or a major trip has just begun. Challenges, openness scientific which break barriers and easy communication between peoples, and the rapid change which appeared on all aspects of life make it the duty of the educational institutions to take different means of modern education. Scientific and technological development have added a lot of new teaching means that can be utilized in the creation of areas of expertise for learners to prepare the individual with a high degree of efficiency to face the challenges of the times (Collis, Nikolova & Martcheva, 1995).

Morning hours of work and social engagements were not an impediment to obtain the academic certificates or stand up and learn about the latest scientific developments in various disciplines, through the offices and centers link to these universities, and provide the learner in a way to the university website, in order to receive a scientific article in anytime and anyplace.

Rapid technological progress contributed the revolutionizing and considered as a big jump in libraries and information science, and significantly enriched the scientific research, which resulted in the emergence of electronic library, databases of electronic information, electronic publishing, e-books and e-journals (which provide the researchers with information in electronic form). Information and communication technologies have been the most dominant influence in our everyday lives (Hajaya, 2010).

Databases, web, and other kinds of repositories which store digitized information such as; image, video, audio and hypertext data, contain a growing amount of audio contents that include newscasts, video tutorials, movies, sporting events, telephone conversations, recordings of conferences or meetings, media files in libraries and others all these are useful information to users, that want to benefit from most of these materials efficiently (implicit knowledge, multimedia data relationships, or other patterns not explicitly stored in multimedia files) by applying indexing and search techniques to these digitized audio contents (Leavitt, 2002).

Although valuable information may be hiding behind the multimedia (video & audio), the overwhelming data volume makes it difficult (if not impossible) for human beings to extract them without powerful tools (such as statistics tools, artificial intelligence and graphs that making by computer), which can automatically extract semantic meaningful information (knowledge) from multimedia files (Kotsiantis, Kanellopoulos & Pintelas, 2005). The traditional analysis methods (statistical, for example) became unable to cope with the vast amount of data, data mining (Knowledge discovery from data or extracting knowledge from large amounts of data) concept emerged as one of the successful solutions to analyze huge amounts of data, and convert it from a mere accumulated knowledge and incomprehensible (data) to valuable information that can be exploited and then utilize them (knowledge). The development of faster and more powerful processors, larger storage capacities, and better speech-recognition algorithms and tools made data mining easier and more efficient (Leavitt, 2002).

Because of the flexibility and continuous progression, education become very adaptive to new technology trends. This means that education (both teaching and learning) can benefit from new IT-based applications that save time and cost and give practical solutions for both teachers and students (Hajaya, 2010).

1.1.1. Audio Mining

Audio mining is using techniques by which the content of an audio material can be (automatic) analyzed, indexed, and searched. It is most commonly used in Automatic Speech Recognition, where the analysis identifies speech within a given audio content. The audio content will typically be processed by a speech recognition system to find words or phoneme units that are likely to be heard and recognized by human (Colbath, Kubala, Liu & Srivastava, 2000).

The result of search is usually measured in terms of hits for parts within audio files that represent good matches for either the searched by keyword, synonym and meaning of a keyword, or query by voice. The user can listen to these parts (hits) in order to verify if these parts represent the correct match between the searched keyword and the found region within the audio file (Leavitt, 2002).

Audio mining, which is also called audio searching, takes a text-based query and locates the corresponding part of an audio content that has the search phrase. For example, a user can quickly get to the exact parts of an audio content to decide when the name of a politician or a country is mentioned in a newscast.

1.1.2. Audio mining approaches

Leavitt (2002) stated that there are two main approaches to audio mining:

1. Text-based indexing: Text-based indexing, which is also known as Large-Vocabulary Continuous Speech Recognition (LVCSR), is a technique which converts speech to text and then recognizes words using a dictionary which is usually built-in within the recognition tool. This dictionary can contain several hundreds of thousands of entries. If a word or a name is not found in the dictionary, the LVCSR system will choose the most similar word to the one it recognized. The system uses language understanding to give results with a reasonable level of confidence (Leavitt, 2002).

2. Phoneme-based indexing: Phoneme-based indexing does not convert speech to text as it only works with sound (phonemes). The system analyzes a piece of audio content to recognize and find sound, and then it creates a phonetic-based index, then it uses a dictionary that has several dozens of phonemes to convert user's search term to the correct phoneme string, (Phonemes are the smallest unit of speech in a language, a word is a set of phonemes). Finally, the system looks for the search term in the index. Phoneme-based approach is less accurate than the text-based approach, particularly for short search terms, because many words sound alike and some words sound like some parts of other words (Leavitt, 2002).

In this study, we highlighted on a new technique aimed to build an intelligent search system for smart lecture archiving by and broaden the search to include large search base within audio and video content depends on text-based indexing for both keyword with their synonyms and query by voice as an alternative method too. We use the term "**Multi-Search**" which means a search by query based on voice query, and text query (keywords with their Synonyms).

1.2. Problem Definition

Most research works in searching and retrieving video content processes depend on traditional methods (manual tracking), searching for keywords which depends match style (text-based indexing), searching web as a searching engines, and semantic search (special high level semantic concepts with relationships), there is no research that combines all aspect of these methods together as a hybrid method.

The development of faster and more powerful processors, larger storage capacities, and better speech-recognition algorithms and tools, made audio mining easier and more efficient, which may contribute in facilitate search and retrieval, hence, increased attention on the side of audio content.

As a result, there are many studies that discuss and suggest sub solutions, from this place we pay most attention to this issue to improve and enhance the process of video searching and retrieving through combines these methods together, expanding and improving the search base by adopting voice query and keywords with their synonyms in searching process (Hybrid System).

This study will answer the following questions:

1. Does search within the video content by keywords and their synonyms will improve retrieved results?
2. Does using voice query as an alternative to text query will facilitate the searching process for video content?
3. Does this system will contribute in raising the level of self-education for learners?
4. Is our approach useful to be used ?

1.3. Objectives

This study seeks to achieve the following objectives:

1. Replace the traditional methods depending on keyword search with a searching ways that depend on the search for the keyword and their synonyms.
2. using voice query as an alternative to text query to facilitate the searching process for video content?
3. Overcome the search difficulties within the audio content by expand searching base.

4. Retrieving appropriate parts of audio content for the query quickly and conveniently.

1.4. Motivation

The significance of this study stands out to help university students and scientific institutions in searching and retrieval of the audio content, by improving the method of search, and expands it to accommodate as much as non-specific keywords unlike traditional search method. Although develop a new way based on the query voice instead of the query text, this will lead to the improvement in the effectiveness of the search for the most accurate information toward a successful search audio content.

1.5. Limitation

When we search for entity that do not have synonyms like names the results might be narrow.

The results that we can get from the voice query may be low due to several factors among them the inaccuracy of the speaker's voice (The degree of purity of the speaker's voice and the pronunciation), lack of clarity of the sound, and the noise around speaker (The surrounding noise). So there are preparatory actions needed before the search process.

1.6. Terminology

There are many terms that have been used through this thesis, in the following are the description of the most common terms:

1. **Multimedia:**

 - The use of text, graphics, animation, pictures, video, and sound to present information (Nahhar, 1996).
 - Refer to the combination of text and images on a computer display terminal (Geoffrey & Mactavish, 2007).

Terminology procedural for multimedia is a set of element include text, sound, video, images and animation which makes computer smart interactive environment and easy to use.

2. **Information:**

 - Refers to collection of facts organized in such a way that they have extra value beyond the value of the facts themselves (Stair, Moisiadis, Genrich & Reynolds, 2011).
 - Refers to data placed in a context and acquires meaning (Cooper, 2014).

Terminology procedural for information is facts and data that change the cognitive status of a person in a particular subject.

3. **Data**:

Data is a set of raw facts/materials that can be recorded and have implicit meaning. The system uses the data called data processing system such as computer system which is a digital data processing system (Stair, Moisiadis, Genrich & Reynolds, 2011).

4. **Database**:

A database is a collection of related data with an implicit meaning. database can be considered as a warehouse which includes and contains all the information about an organization or a company... etc, whatever the size of the organization or the type of information (Elmasri & Navathe, 2010).

5. **E-learning**:

The letter "E" in E-Learning stands for the word "electronic". E-Learning means the usage of available information on networks whether local or extended one and communication media in synchronous or asynchronously teaching and learning activities (Naidu, 2003).

6. **Digital Lecture:**

Any lecture is being ditched by using the technology of digital information across networks services whether local or widely (Demetriadis & Pombortsis, 2007).

7. Data Mining:

It is the process of mining and extracting knowledge from large amounts of data and also called "knowledge mining from data" (Han & Kamber, 2006).

8. Speech recognition:

Known also as "Automatic Speech Recognition (ASR)" and defined as the process of converting speech signals and languages spoken to a series of words or readable texts using special software based on recognize speech algorithms (Anusuya & Katti, 2010).

9. Audio:

Audio is a sound within the acoustic range available to humans, audio in computer refers to the sound system that comes with or can be added to a computer [1].

10. Video:

Video refers to mixture of audio/visual (animations with voice) in a binary format and represented in a continuous sequence of digital or analog data [2].

1 http://whatis.techtarget.com/definition/audio, 2014
2 http://searchsoa.techtarget.com/definition/digital-video , 2014

1.7. Thesis Structure

This thesis consists of six chapters: Chapter one: Introduction, problem definition, objectives, importance, limitation and some terminology of the research. Chapter two: Literature review (comprehensive literature survey in audio mining and video retrieval) and some related works for our work. Chapter three: Methodology and main component of our system. Chapter four: The proposed prototype system and a complete description of the working procedures. Chapter five: Results and comparison with nearly work, and finally chapter six: Conclusions and future work.

CHAPTER TWO

Literature Review and Related Work

Literature Review

2.1. Introduction

Information and communication technologies are dominating our everyday lives; Education, social communication, entertainment, business, health care, military training and techniques are all affected by rapid technical evolution. The educational aspect is one of the aspects that most fit with new technology, thus benefit from new IT-based applications that save time, effort and cost.

2.2. The Educational Process

Teaching practices centered on individual educational situations, and increased degree of freedom given to students in learning situations, are increased the options and the alternatives to them. Technological innovations have contributed effectively to the emergence of many other educational applications such as computer-assisted learning, internet learning resource centers, multimedia technology, and open universities. Using methods to learn more widespread will guarantee access to a greater range of learners, but it does not mean the quality of the interaction of this medium with these learners. As such, it refers to the quality of education, it is not measured only with a capacity of spread, but its ability to interact with learners and meet their different needs (Collis, et al., 1995).

The real purpose of the Internet in the educational process is to give the learner more control in learning, in terms of the possibility of his arrival to the information depending on individual initiative and abilities in research and access to information, it led to the need for an alternative way to make decisions. Learning through the Internet became a new vision for learning to look beyond the textbook, or method of memorization; it has several features marked advantage for the rest of the forms of learning [3].

Using the computers in education through quality educational software provide the student with a tremendous momentum of real interaction during the learning process, outweigh any other educational way.

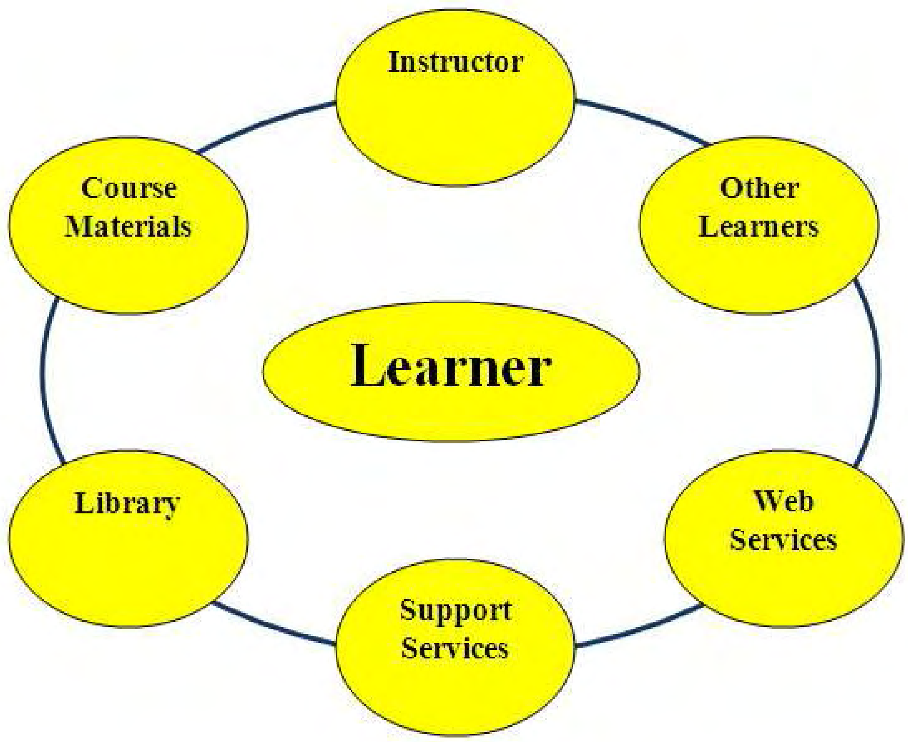
2.3. E-Learning

E-Learning is any technology-supported learning initiative, or it is the combination of educational content of service and learning experiences with electronic technology. It uses the information and the communication technology (ICT) in education to support and enhance learning process. The transition from a traditional learning style to e-learning style may lead to some changes in the culture and the development of the skills of

3 http://www.edudemic.com/the-importance-of-internet-access-in-schools, 2014

employees and students in technology skills (Aladwan & Smedley, 2012). Nowadays e-learning is used as a supplement to traditional classroom due to one-content-fits-all approach of teaching and learning. The most important element in the educational process is the learner who consider as the center of the learning system as shown in the following figure:

Figure 1: E-learning environment



Source: Hajaya (2010)

E-learning offers several important services:

- On-demand availability: Information is available on demand.
- 24\7\365 accessibility: E-learning resources are accessible (global access) at anytime from anywhere.

- Flexibility: Enabling learners to engage in the learning process anytime, anyplace and on a just-in-time basis.
- Decreased cost of learning delivery, reduced travel, subsistence costs and time away from the job.
- Increases the communication and collaboration among all participants in the educational process, sharing ideas and viewpoints.
- Facilitating learning opportunities for aboriginal students and those in more remote areas.

2.3.1. Digital Lectures

The widespread usage of e-learning systems and their evolution has led to become part of the educational process in universities and training institution, which focuses on providing course content (E-Lectures) in an electronic format, such as view educational material electronically communications, e-registration and online exams, but there are many negative aspects of using E-Lecture system and more of these drawbacks is the need for special tools as intermediates needed to download some players to be able to view the E-Lecture, also need to coordinate, and to teach people how to compose a short lecture in the required format (Hirzallah, 2007).

Web-based learning systems are the most alternative methods to traditional in-class lectures (Erol & Li, 2005). Today with the evolution in multimedia technologies, current E-Lecture systems have gone far beyond the stage of plain video recording and display, it became involved capture, analysis and recognition, which led to the creating new ways to get access to, search and analyze both real-time and archived meetings.

Availability of equipment needed to create E-Lecture such as cameras with low prices and easy way to install it, the editing software, the advanced computers with high storage capabilities are maximize role of multimedia in education, and motivated many of the recent technological developments in producing, storing, and using of audio / visual materials with acceptable cost and effort.

2.4. Data Mining

Data Mining is one of the most important area in information technology due to the availability of huge amounts of data, that we need to turn it into useful information and knowledge. The advantage of this and the resulted knowledge can be used in several areas, especially in the field of the applications, such as e-learning (Cao, 2010). Users must be able to extract knowledge from easily available materials, by using the data mining techniques.

Han & Kamber (2006) stated that data mining refers to extracting or "mining" knowledge from large amounts of data, sometimes, data mining is considered as an essential step in Knowledge Discovery from Data (KDD).

KDD consists of an interactive sequence of several steps:

1. Data cleaning (to remove noise and inconsistent data).
2. Data integration (where multiple data sources may be joined).
3. Data selection (where data relevant to the analysis task are retrieved from the database).
4. Data transformation (where data are transformed or consolidated into forms of mining by performing summary or aggregation operations).
5. Data mining (an essential process where intelligent methods are applied to extract data patterns).
6. Pattern evaluation (to find the truly interesting patterns representing knowledge based on some interesting measures).
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

The above data reprocessing steps are described in the following figure: (Han & Kamber, 2006)

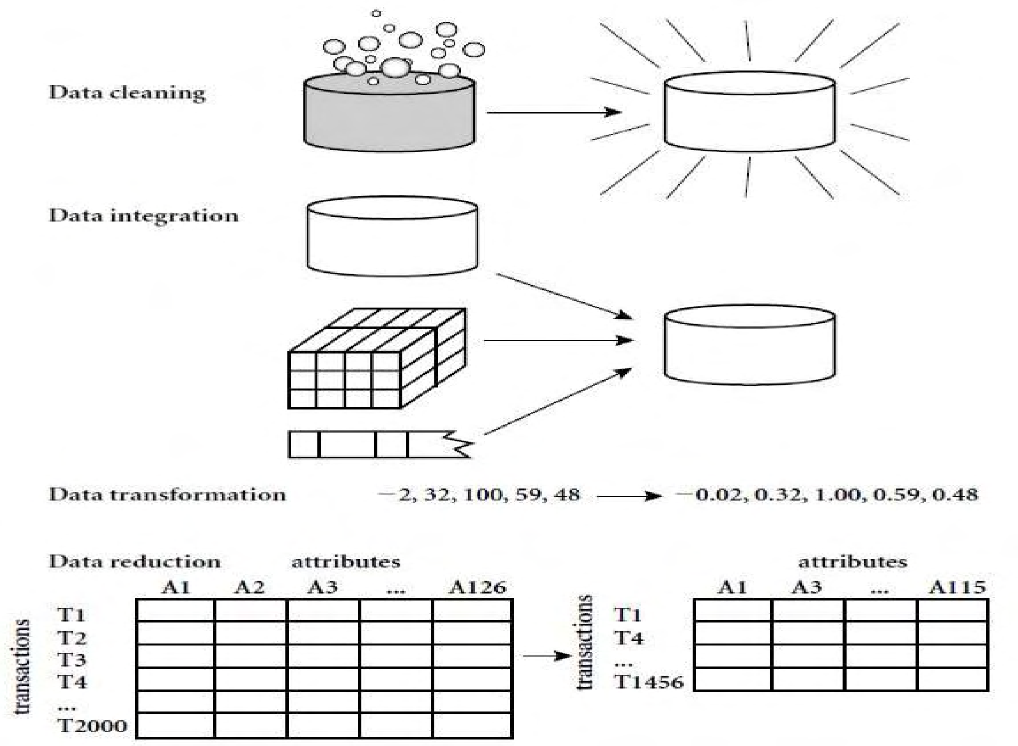


Figure 2: Forms of data preprocessing (Han & Kamber, 2006).

2.4.1. Multimedia Data Mining

Han & Kamber (2006) stated that multimedia data, such as audio, video, image, graphics, speech, text, web pages, office documents, technical drawings, 3-D Computer-Aided Design (CAD), and hypertext data are used in learning materials and became widespread, so if these multimedia files are analyzed, useful information to users can be revealed, such as help recording, search, and analysis of audio and video information from multimedia data, then allow users to use those materials in best way. In searching multimedia data specially for similarity searching in multimedia data, there are two main families of multimedia indexing and retrieval systems are used; these are:-

1- The description-based retrieval systems based on image descriptions, such as keywords, captions, size, time of creation.
2- The content-based retrieval systems based on the image content such as color histogram, texture, pattern, and image topology.

The following figure summarizes the multimedia mining process, initially mining system begin collecting raw data which greatly affect the overall performance of the system . Then, discovering important feature of raw data through data processing, which includes as already mentioned above, data cleaning, transformation, feature selection. The outputs of data pre-processing is the training set on which learning model has to be chosen to learn from it.

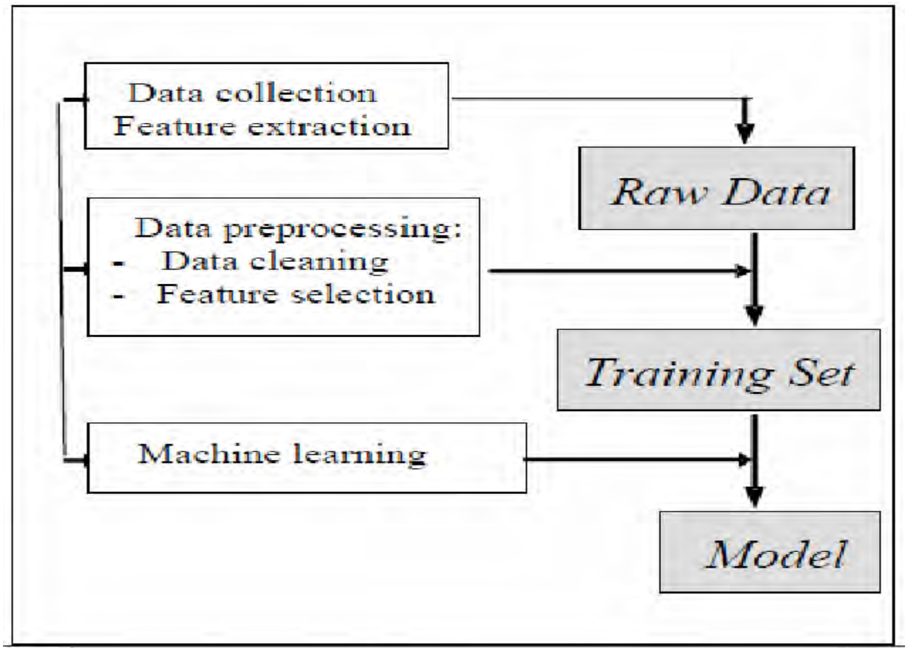


Figure 3: Multimedia mining process (Han & Kamber, 2006).

2.4.2. Audio Mining

Audio mining which sometimes called audio searching takes a text-based query and locates the search keyword or phrase in an audio file (Han & Kamber, 2006). Audio mining can be used in many fields such as companies as a customer service, voice mail and in law fields as law enforcement, gathering evidences, uncovering the facts and in the military field especially in the field of electronic reconnaissance through tracking, analyzing and intercepting phone conversations to decide whether it is dangerous as terrorist or not, or to find illegal activity. On the other hand in the field of sports, this technique is used in analyzing the match, in radio stations, they use audio mining to retrieve information from earlier broadcasts, however this technique can be used in many areas of life, in the light of the above, this is the reason for manufacturing and producing of many audio mining software.

Audio indexing techniques use speech recognition methods in the process of analyzing files and producing a searchable index of content identifies words and locations, this is very important because audio content is usually in a binary format which is not a searchable file format (Leavitt, 2002).

The following figure shows the flow chart of an audio mining system.

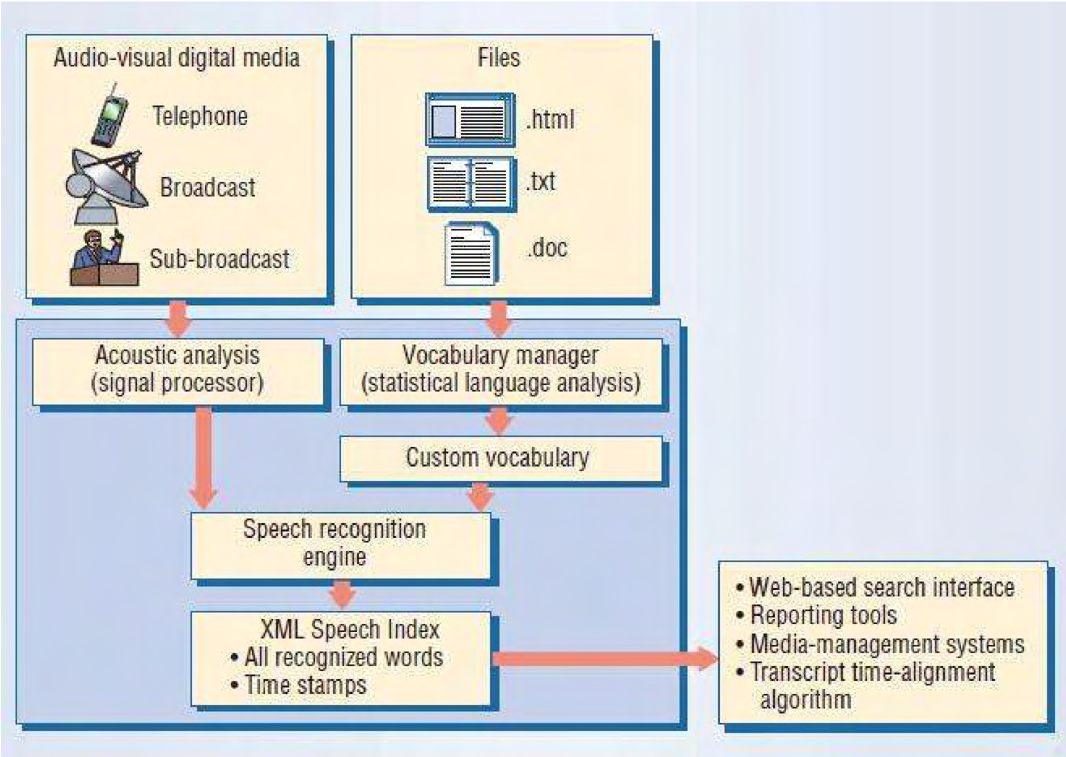


Figure 4: The flowchart of a general audio mining system (Leavitt, 2002).

The system analysis of audio content to produce sequential parts of the sound, which can be switched to the words through speech recognition engines, depends on its own dictionary. Material input into the vocabulary manager updates the dictionary, and then the produced Extensible Markup Language (XML) Speech Index can be used with various search engines, servers, and other content management systems.

In this proposed system, Text-based indexing is used instead of Phoneme-based indexing because it provides search results with more accurate matches. Phoneme-based indexing systems may still be useful when searching materials that contain words that are unlikely to be found in system's dictionary, such as foreign terms, names of people and places. Those terms are unlikely to be found in video lectures. The difference between those two systems that text-based system uses a text-based dictionary and phoneme-based system uses a phonetic dictionary (Leavitt, 2002).

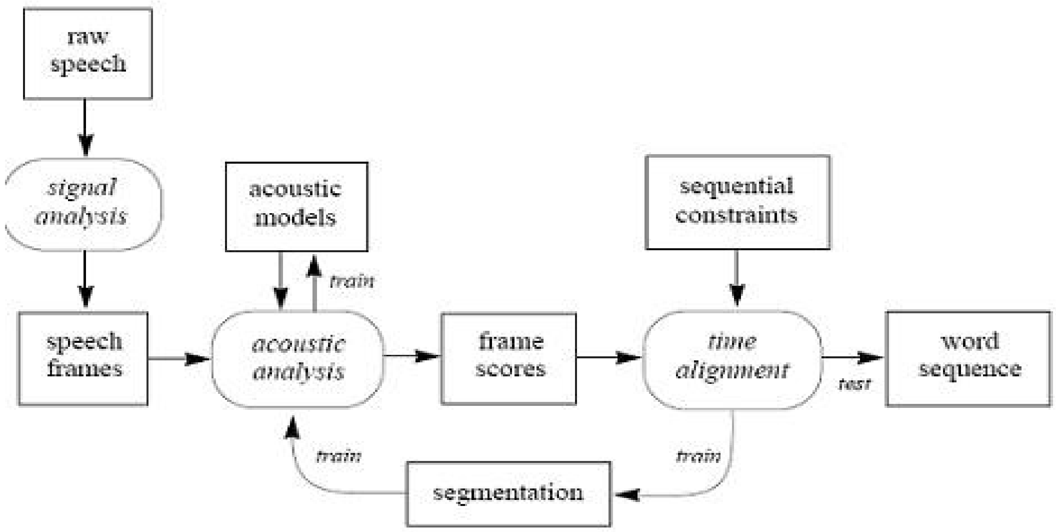
2.5. Speech Recognition

Speech recognition is the process of reconstructing the text of a spoken sentence from the continuous acoustic signal induced by the associated utterance (Buchsbaum & Giancarlo, 1997). Sometimes defined as the process of converting an acoustic sound from microphone or telephone into words. The speech recognition process is performed by a software component known as the speech recognition engine. The primary function of the speech recognition engine is to process input spoken and translate it into text (Hajaya, 2010). Nowadays there are many Automatic Speech Recognition (ASR) systems that recognize the speech in real time. The accuracy of speech recognition systems depends on pronounce words properly, origin of the word, the linguistic background of the speaker, and other cultural and sociological factors, in addition to the word spelling (Beaufays, Sankar, Williams & Weintraub, 2003).

Vrinda & Shekhar (2013) stated that there are several definitions and concepts that must be understood before understand the big picture of Automatic Speech Recognition (ASR):

- **Utterance**: An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences. Utterances can also be continuous (spoken without silences) or discrete (consists of isolated words that are separated by silences).
- **Speaker Dependency**: Speaker dependent systems are designed around a specific speaker. They are generally more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo. Some speaker-dependent systems require only that the user record a subset of system vocabulary to make the entire vocabulary recognizable, speaker independent systems are designed for a variety of speakers, it does not require any recording (for example record the word, sentence, or phrase system use). Adaptive systems usually start as speaker independent systems and utilize training techniques to adapt to the speaker to increase their recognition accuracy. The speech recognition system is a speaker independent system.

- **Vocabularies**: Vocabularies (or dictionaries) are lists of words or utterances that have to be recognized by the ASR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult.
- **Accuracy**: The ability of a recognizer can be examined by measuring its accuracy (word error rate), or how well it recognizes utterances. The accuracy varies from speaker to speaker.
- **Training**: Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match the particular speaker. Training a recognizer usually improves its accuracy. The structure of a standard speech recognition system is illustrated in Figure 5 [4]:



4 http://www.learnartificialneuralnetworks.com/speechrecognition.html, 2014

2.5.1. Types of Speech Recognition

Anusuya & Katti (2010) stated that speech recognition systems can be classified into several different classes by describing what types of utterances they have the ability to recognize. These classes are based on the ability of ASR system to determine when a speaker starts and finishes an utterance These classes are classified as the following:

1. **Isolated Words**: Isolated word recognizers usually require each utterance to have quiet (lack of an audio signal) on both sides of the sample window. It accepts single words or single utterance at a time. These systems have "Listen/Not-Listen" states, where they require the speaker to wait between utterances (usually doing processing during the pauses). Isolated Utterance might be a better name for this class.
2. **Connected Words**: It is similar to isolated words, but recognizers allow separate utterances to be 'run-together' with a minimal pause between them. Connected Utterances might be a better name for this class.
3. **Continuous Speech**: In this class recognizers allow users to speak almost naturally while the computer determines the content. Basically, sometimes known as computer dictation.

4. **Spontaneous Speech**: There appears to be a variety of definitions for what spontaneous speech actually is. At a basic level it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together "ums" and "ahs" and even slight stutters.

2.5.2. Speech Recognition process

Hajaya (2010) argued that most recognizer can be broken down into the following steps:

1. Audio recording and Utterance detection: can be accomplished in a number of ways; first: comparing ambient audio levels (acoustic energy in some cases) with the sample just recorded. Second: detection .
2. Pre-filtering (pre-emphasis, normalization, banding, etc.): done by a variety of ways, the most common method is the "Bank-of-Filters" which utilizes a series of audio filters to prepare the sample.
3. Framing and Windowing (chopping the data into a usable format): involves separating the sample data into specific sizes.
4. Filtering (further filtering of each window/frame/Freq Band): additional filtering consider the final preparation for each window

before comparison and matching, often this consists of time alignment and normalization.

5. Comparison and Matching (recognizing the utterance): can be performed by comparing the current window with known samples. There are methods that use Hidden Markov Models (HMM), frequency analysis, differential analysis, linear algebra techniques/shortcuts, spectral distortion, and time distortion methods. All these methods are used to generate a probability and accuracy match..
6. Action (Perform function associated with the recognized pattern): can be just about anything the developer wants, for example, writing the result text into a file or performing the result text as a command.

Although each step seems simple, each one can involve a multitude of different (and sometimes completely opposite) techniques.

2.6. Knowledge Base System (KBS)

Knowledge-based systems are Systems based on knowledge and depend on it. The analysis of the knowledge based system depending on the proposed functional model of human system which are built according to the direction of arrow from top to down, but the design and implementation of knowledge based system will be from bottom to up, but with some small different in human system the communication with the environment via vision and the long term memory will be user interface in knowledge-base system and knowledge-base respectively (Owaied & Abu-Arr'a, 2007). The following figure presents the construction of KBS.

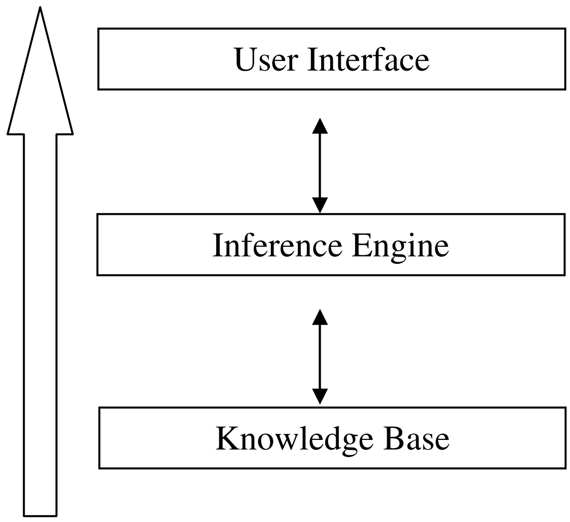


Figure 6: Construction of KBS

The inference engine consists of the following non touchable activities which are: willing and needs, vision, Incentives, hobbies, and the effect of environment. Therefore, no human can have any job without having the knowledge of doing that job, and the knowledge can't be well represented in his long term memory without the influence of the combination of the activities. Human solving problems usually start from the communications with the environment by gathering information assertion about the problem, then the inference engine call the prior knowledge existed knowledge in long term memory associative (Owaied & Abu-Arr'a, 2007). The following figure explains the implementation model of human system.

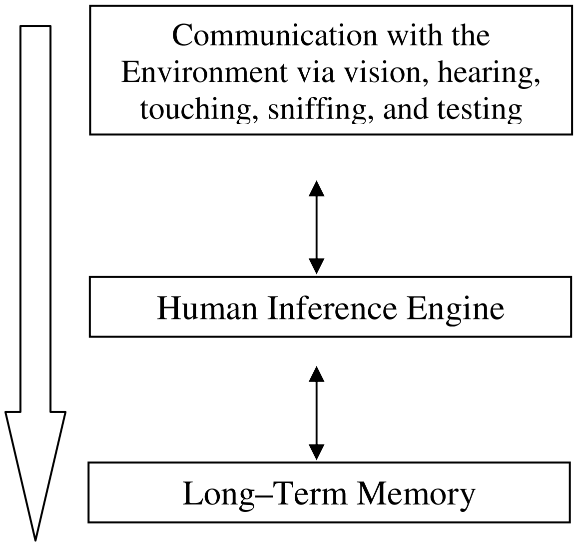


Figure 7: The implementation model of human system

The direction of arrow presents the implementation of model of human system .The implementation of knowledge depends on forms of representation of the knowledge (Rule base, Case base, Frame and Script, Semantic nets, Logic forms, Case base, Model base and other), and the implementation of the knowledge based system as a problem solving method, search technique, and reasoning agent. The user interface may use Question and Answer, Menu-driven, Natural language, or graphics interface styles (Owaied & Abu-Arr'a, 2007).

2.6.1. Expert System

The expert system is a Knowledge based system with ability of interpretation, conclusion and take the correct decision (Lucas & Gaag, 1991). The structure of expert system design is the follow:

Expert System = Knowledge + Inference.

The following figure illustrates expert system part.

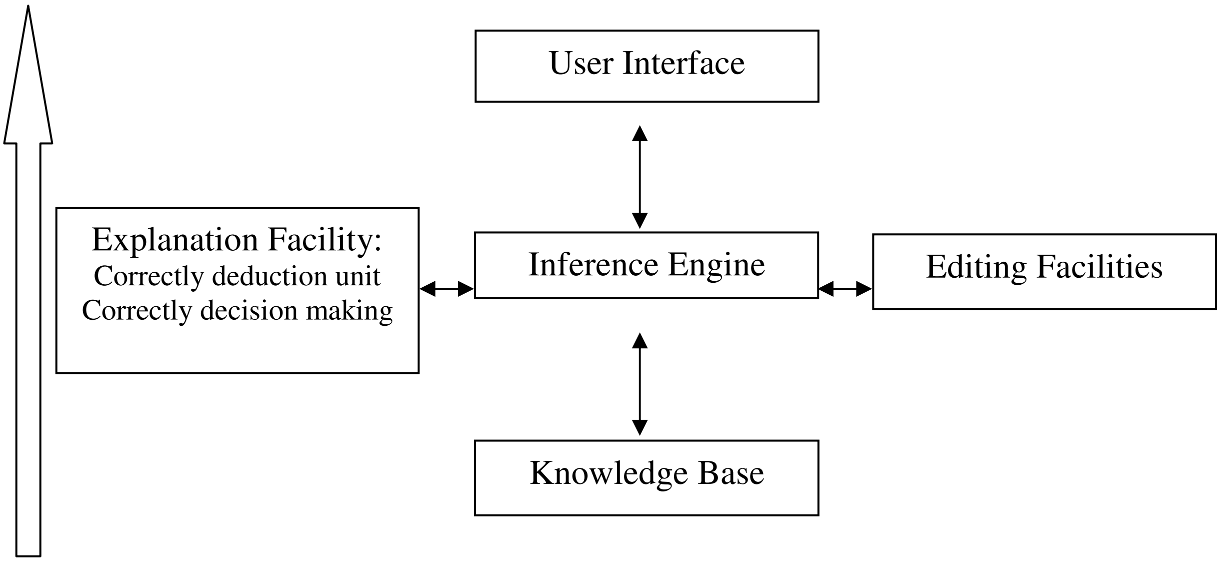


Figure 8: Global architecture of an Expert System (Owaied & Abu-Arr'a, 2007).

2.7. Information Retrieval Systems

Information retrieval systems is one of the subsystems in the information system that is capable of storage, retrieval, and maintenance of information according to users' needs, whether it is inside or outside the library. Information retrieval is just a mean or tool that the researcher and the users use it to reach to bowl of the information that is looking for. The information stored by the computer in the mail or bank information system is text (including numeric and date data), images, audio, video, and limited information on each pot, such as address, date the responsible for its content and intellectual publication, number of pages, or leaves, and the heads of the subjects (Kowalski, 1997).

The main purpose of an Information Retrieval System is to minimize the overhead (that expressed as the time a user spends in all of the steps leading to reading an item containing the needed information for examples; query generation, query execution, scanning results of query to select items to read, reading non-relevant items) of a user locating needed information.

2.8. Search Techniques

It is important to identify the scope of task in terms of the problems which need to be solved, for instance, there are some tasks which are single problems solved by searching, e.g., find a route on a map. Alternatively, there are tasks such as winning at chess, which have to be broken down into sub-problems (searching for the best move at each stage), other tasks can be achieved without searching whatsoever e.g., multiplying two large numbers together.

One of the most important things that should be considered is the speed of search because of its significant impact on the cost and time of the resulting path. Efforts to solve problems with computers which humans can routinely solve by employing innate cognitive abilities, pattern recognition, perception and experience, invariably must turn to considerations of search (Gonzalez, Diaz-Herrera & Tucker, 2014).

Gonzalez, Diaz-Herrera & Tucker (2014) stated that all search methods essentially fall into one of two categories:

1. Exhaustive /Uninformed /blind /Brute-Force search methods: is a class of general purpose search algorithms that operate in a brute-force way, these algorithms can be applied to a variety of search problems, but since they don't take into account the target problem. It has no information about the number of steps or the path cost from the current state to the goal, such as Breadth-first search, Depth-first search.
2. Heuristic /Informed /Directed search methods: an informed method of searching a state space with the purpose of reducing its size and finding one or more suitable goal states by using information about the domain to (try to) usually head in the general direction of the goal node(s), such as Greedy search, Genetic algorithms.

Heuristics play a major role in search strategies because of exponential nature of the most problems. Heuristics help to reduce the number of alternatives from an exponential number to a polynomial number. The solution may not be the best of all the actual solutions to a given problem, or it may simply approximate the exact solution, but it is still valuable.

2.9. String Matching

String matching means finding one or all the occurrences of a pattern in a larger body of text (e.g., a sentence, a paragraph, a book, etc.). The pattern and the text are both strings built over a finite alphabet (a finite set of symbols).

2.9.1. Types of String Matching

All string matching algorithms divide into two major types (Gonzalez & Tucker, 2014):

1. Identical String Matching: when one pattern becomes identical to (part of) another pattern. The word exact is often used to denote algorithms that compute an optimal solution.
2. Approximate String Matching: when a pattern becomes similar (but not necessarily identical) to another pattern. An approximation algorithm computes a solution that is not necessarily the optimal one.

Related Work

- Colbath, S. et al. (2000) create searchable archives from continuous audio by introducing a system that produces a ROUGH transcription (ROUGH is audio indexing system), which is ready to be browsed. The system is focused on transcription and indexing of broadcast news. To give structure to the audio stream, the system uses the linguistic information recovered to offer the user with a searchable index. It integrates seven advanced speech and language technologies: speaker segmentation, clustering, identification, speech recognition, name spotting, topic classification, and story segmentation (this part transforming the audio from wave file to indexed database). The main idea of this system is to archive the sound into text and build a structural summary of continuous audio that is searchable by content, as illustrate in figure 9. The drawback of this system lacks dealing with noise associated with the voice.

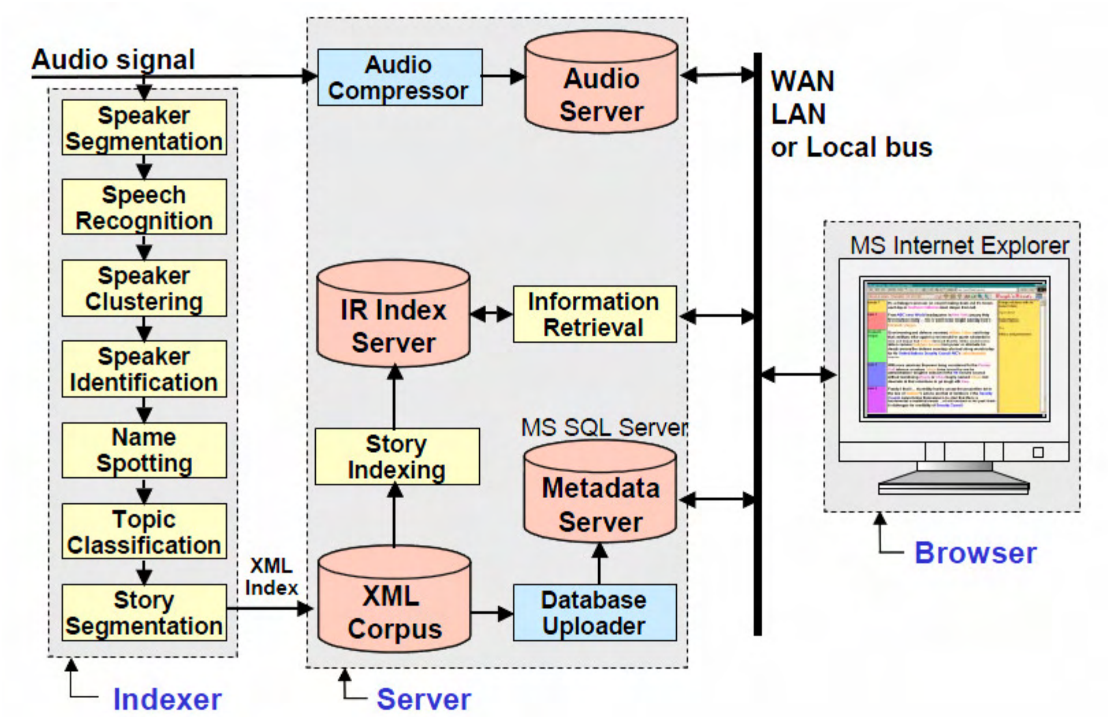


Figure 9: The structure of a standard speech recognition system.

- Coden, R. & Brown, E. (2001) have solved the problem of finding collateral information that is related to a live television broadcast in real time streaming to create a much richer television viewing experience. The solution starts with a text transcript of the broadcast generated by an automatic speech recognition system. To improve the accuracy of speech recognition technology they use an algorithm for their development that can decide the core of the broadcast from these transcripts, where this algorithm extracts named entities, topic, and sentences from the generated transcripts and use them to generate queries. Finally, the query results (collateral information) are then added to the broadcast stream. The main idea is allowing adding relevant collateral information to live programming in real time based on the words spoken during a news broadcast. The benefit of this approach is that over the time more modules can be developed and integrated seamlessly into the system, but drawback of this system is background noise, Slowness and cost. Figure 10 shown the system.

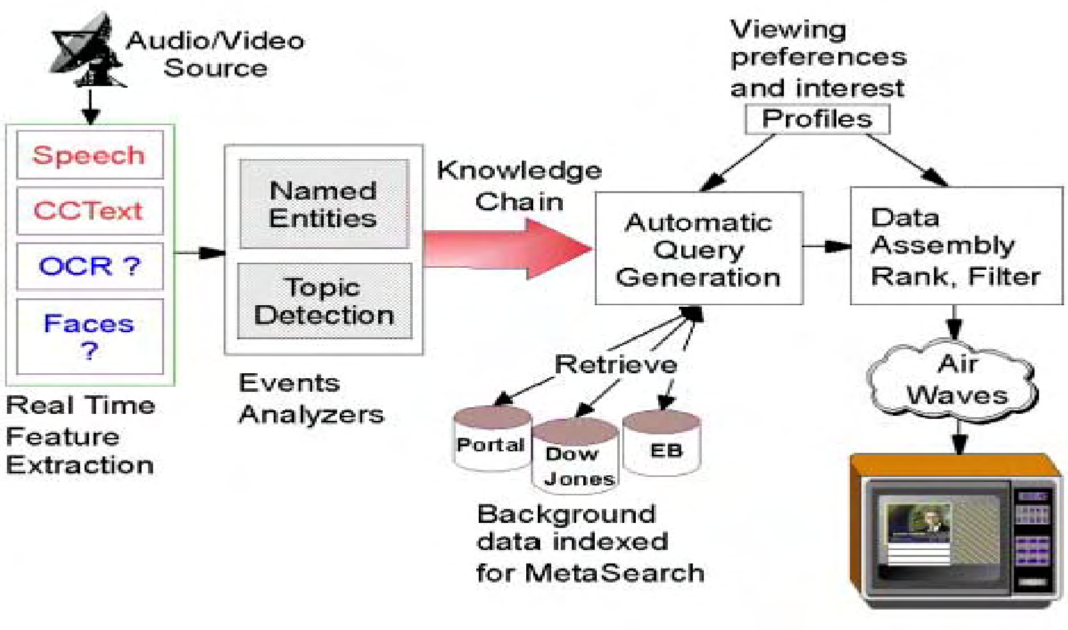


Figure 10: Architecture overview for Coden. R & Brown. E system

- Kawahara. T (2004) Focus on Spoken Language Processing for Audio Archives of Lectures and Panel Discussion. A new system designed to archive lectures and panel discussions based on automatic transcription and indexing, which are the main materials in academic communities. Relied on more than one approach through the process of archiving and indexing of lectures audio. Figure (11) displays an intelligent lecture archiving system submitted by him.

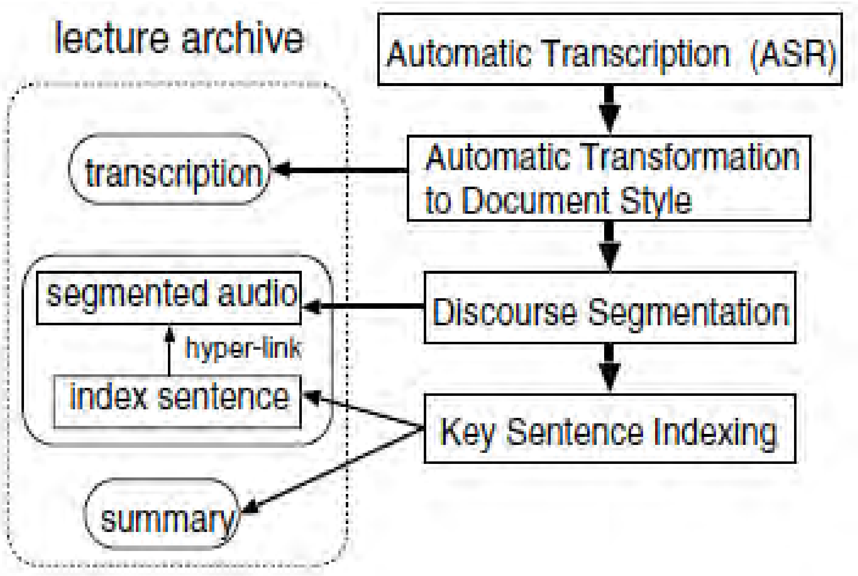


Figure 11. Overview of Kawahara lecture archiving system

As seen in the previous figure, the whole speech is automatically transcribed by an ASR system, the result text is then automatically transformed to document-style sentences for improved readability. Then, the discourse segmentation into section units is performed and key sentences are indexed for each section. Collection of these sentences might also suffice a summary of the talk. In the generated archive, the index sentences are hyper-linked with the segmented audio for easy browsing.

- Haubold, A., et al. (2006) make a system focus on enhance semantic search and retrieval of broadcast news content by applying a set of visual models of semantic concepts from a lexicon of concepts deemed relevant for the collection. Their approach involves a query expansion stage, in which query terms are compared to the visual concepts for which they independently build classifier models. They use TRECVID 2005 broadcast news as a dataset which is limited, they enhance the process of semantic search of multimedia by 50% over text-based search, they apply their solution on 39 concepts specifically designed for this genre of video to validate the approach, figure 12 shows the LSCOM-lite lexicon (concepts):

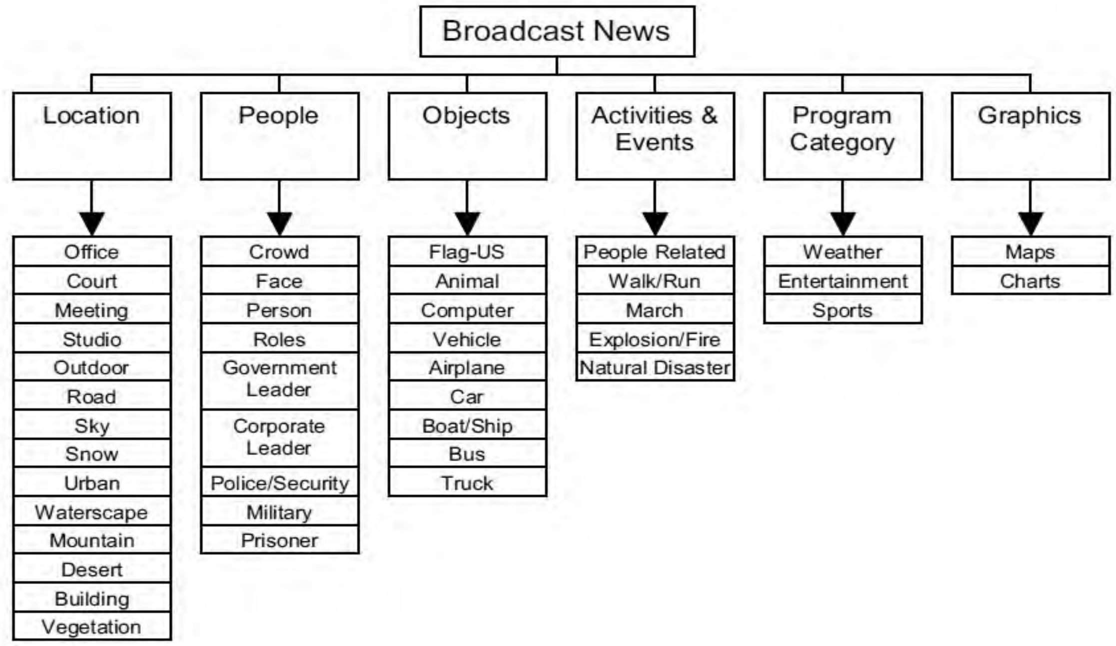


Figure 12. The LSCOM-lite Lexicon (Haubold, 2006)

They use standard and limited test set, we think the test set is not comprehensive and general but it is specific and addresses a specific category. In their approach they worked on classification query by weighting and grouping because they worked in their approach on semantic relationships, while in our approach there is no classification. They calculated concepts score for query terms and calculated concepts score for the shot concepts and built a ranked list of matching shots based purely on query-to-concepts relevance scores. Following figure illustrate the relationship score:

Query: "*people with banners or signs*"	
T1: *people*	*people-marching*: 4737, *crowd*: 4737, …
T2: *banner*	*people-marching*: 216, *US flag*: 151, …
T3: *sign*	*building*: 5361, *waterscape/front*: 531, …
Final query to model mapping with weights: *Building*: 1879.7, *People/marching*: 1685.7, *Crowd*: 1625.3, …	

Figure 13. Example of a complete query expansion with Lesk semantic relatedness scores (Haubold, 2006)

In our approach we did not work on semantic relationships (such as: is a…, has a… relationships) or semantic search because it will adversely affect the results, negative and unexpected results will appear, or perhaps unrequired or undesirable; we did a generic search. The domain of their approach was only on broadcasts news video while our approach deals with generic videos which contain a variety of topics not only broadcasts news video. In their approach they used a video frames already segmented but in our approach we built it manually. Their approach is time consuming and does not scalability if the data set is huge; the complexity of time will increase because huge result will return from searching process and may be contain unrequired results (Using multiple relationships in the search will increase the time complexity).

- Jinjun, W. (2006) worked on Content-Based Sports Video Analysis and Composition. His method is focused on detecting semantic events and event boundaries from both broadcast sports video and non-broadcast sports video and analyzed the webcasting text information associated with the video and synchronized it with the visual/audio features. This method just based on webcasting text information, and webcasting text or other textual information is not universally available.

- Zheng, W., et al. (2006) Worked closely to Haubold, A. work. They present a method called pointwise mutual information weighted scheme(PMIWS) make judgment of the relevance of all the semantic features to the queries. Their experimental results are based on TRECVID2005 corpus. Their method was used to bridge the semantic gap between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation. In their approach they worked on classification and feature weights because they had worked in their approach on semantic which it depends on many relationships, while in our approach there is no classification.

In their approach, they modeled and extracted 33 high-level features which we think is too small and does not cover all the topics of different videos, and then they presented a method called pointwise mutual information weighted scheme (PMIWS) to utilize the information provided by these features. They modeled a detector for the semantic feature and used it to produce confidence scores they used the LSCOM-lite Lexicon[5] too. They calculate the probability of existing semantic feature set (that extracted according to query) in a dataset of video shots depending on the weight and the degree of the relevance between the semantic feature and the concept of shots. Their approach depended on high-level semantic features and concepts of shots, for example (Military concept: mean shots depicting the military personnel). They stated that High-level semantic features are considered unuseful except for a few topics which have well-performing correlated semantic features, and semantic feature sets are used for only (5%) of the interactions of interactive retrieval and contribute to negligible improvement.

5 http://www.ee.columbia.edu/ln/dvmm/lscom/ibmtr2005-lscom.pdf, 2014

Figure below show and illustrate their video retrieval system:

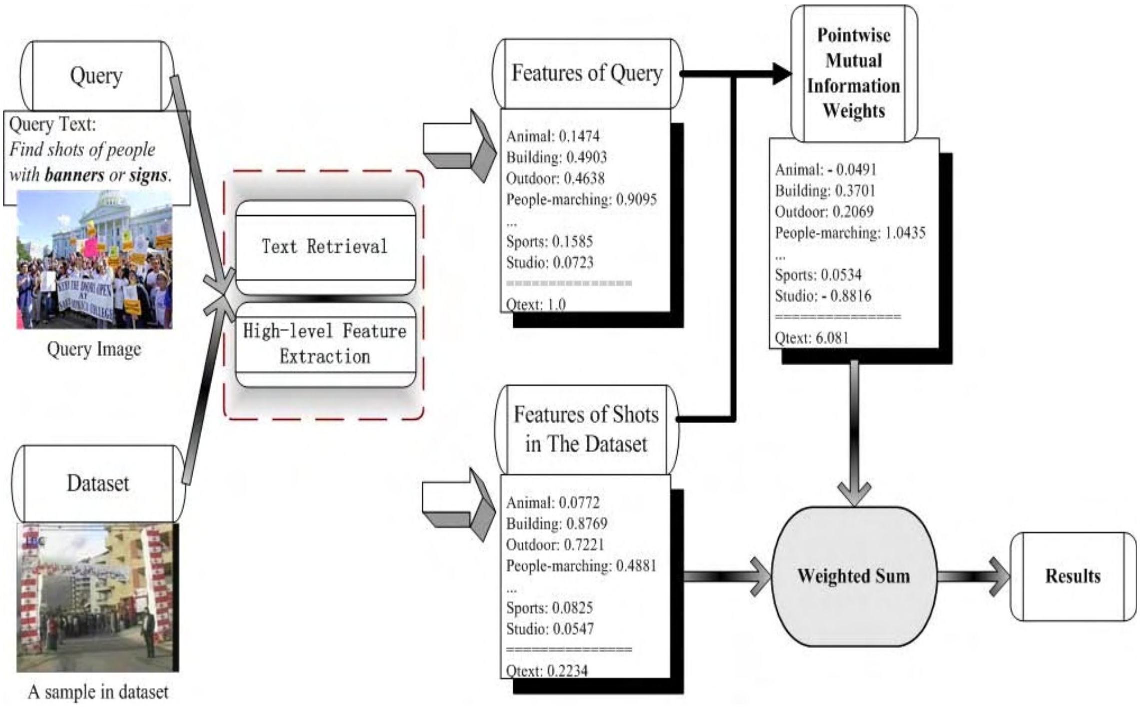


Figure 14. The overview of Zheng video retrieval system

We see that the feature extraction results in their method are not always consistent. They did not analyze text of the query but they took texts of the query as crude. In our approach we did not work on semantic relationships (is a…, has a… relationship) in searching process because it will adversely affect the results, negative and unexpected results will appear, or perhaps not required or desirable; we do generic search. The shots in their approach were classified into concept groups but in our approach the set of shots are generic and not classified. Their approach is time consuming and does not scalability if the data set are huge; the complexity of time will increase because they used semantic relationships and huge result will be return from searching process and may be contain results not required (Use many relationships in search will increase the time complexity).

- Maji, S., & Bajcsy, R. (2007) propose a way of combining weakly associated video/audio and text steams in an unsupervised manner, by aligns audio/video and text streams technique. They have shown that their technique can be used to annotate individuals appearing in the video, with their names. They tested their technique on a 80 minute video segment downloaded from the website of the International Court of the Former Yugoslavia, with the corresponding transcripts. Their proposed technique achieved (88.49%) accuracy on sentence level alignments and (95.5%) accuracy on the task of assigning names to faces. The figure below show the flowchart of their technique:

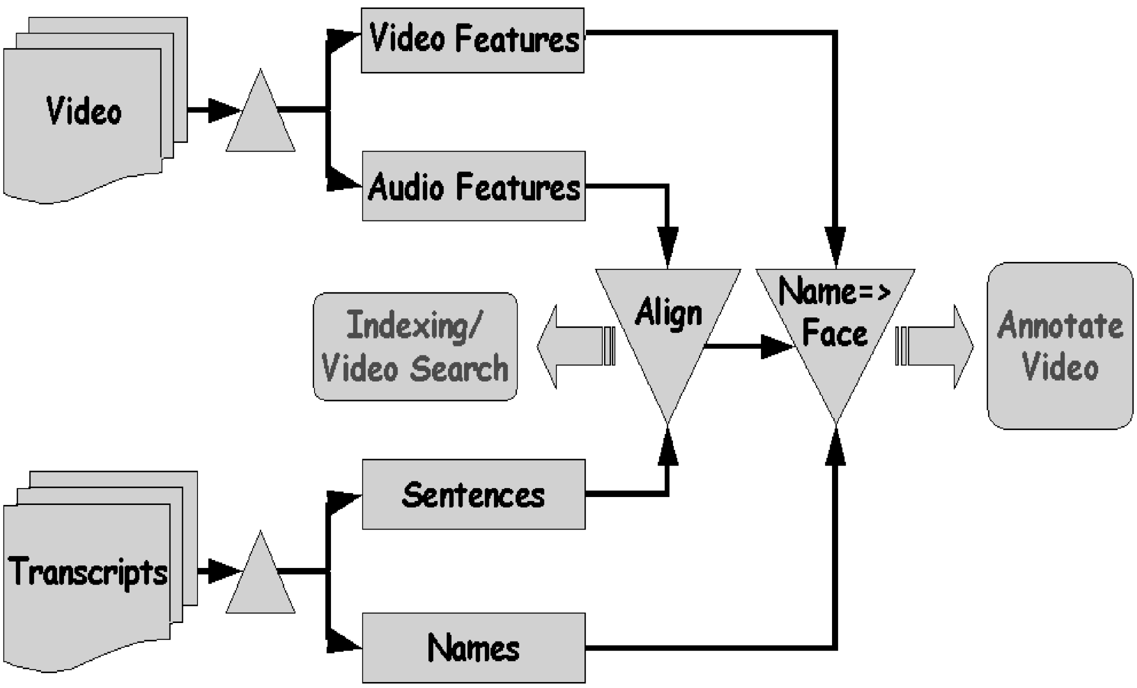


Figure 15. Flowchart describing Maji & Bajcsy technique

- Zhang, Y., et al. (2007) worked on Semantic event extraction from basketball games using multi-model analysis based on webcasting text and broadcast video. Their approach extracts event semantics and use them to annotate and index events in the video by using a new external resource, and webcasting texts. As a result of text analysis, map the text event time-tag into the game video stream can be done to detect event moment and event boundary in the video.

- Lee, et al. (2008) Submitted a prototype design for video search engine called LeeDeo (LEarning and Educational Video Search Engine) that can search, crawl, archive, index, and browse “academic” or "educational” videos from the Web. Those videos include lectures, tutorials, demonstration videos, conference and workshop presentation videos, colloquium videos, and educational User-Created Contents (UCC) videos. The proposed design differs from the well-known search engines like Google and MSN whose focus is on the search of textual HTML documents or metadata of multimedia objects, and differs from existing academic bibliographic search engines such as CiteSeer, arXiv, or Google Scholar whose focus is on the search and analysis of PDF or PS documents of academic papers. It is a special academic video search engine.

The main steps for the proposed LeeDeo system is crawling only academic or educational video clips from the web by utilizing the hyperlinks on the web, classification academic videos from the rest, extraction metadata and transcripts from the classified videos, indexing for search engines, and provide interface for efficient browse and search of academic videos, LeeDeo extracts metadata from videos themselves as well as from web pages where those videos are hosted, it also extracts transcripts from audio tracks of videos using ASR tool and uses those transcripts for advanced indexing.

- Xu, M., et al. (2008) Focused on sports videos, especially on the semantic event detection area in sports, that's where video and audio play an important role in the semantic event detection. He introduced the concept of Audio Keywords which represent mid-level features and used for bridging the gap between low-level features and high-level semantics, and used it to define a set of specific audio sounds.

They put the framework for event detection in sports videos, which is generic and can be applied to any game, especially for ball-type sports video as shown in the table below.

Table 1: Audio Sounds Relationship to Semantic Events.

Sports	Audio Sounds	Potential Events
Tennis	Applause	Score
	Commentator Speech	At the end (or the beginning) of a point
	Silence	Within a point
	Hitting Ball	Serve, Ace or Return
Soccer	Long-whistling	Start a free kick, penalty kick, or corner kick, game start or end, offside
	Double- whistling	Foul
	Multi- whistling	Referee reminding
	Excited commentator speech or excited audience sound	Goal or Shot
	Plain commentator speech or plain audience sound	Normal
Basketball	Whistling	Fault
	Ball hitting backboard or basket	Shot
	Excited commentator speech or excited audience sounds	Fast break, Drive or Score

Sounds contained within sports video give some indication and some potential semantic linkages from audio sounds to interesting events. As shown in Table (1) to detect potential events, some sounds can be noticed like the sounds from whistling, commentator speech, audience sound, and game-specific sounds, such as the ball being hit in tennis or the ball hitting the backboard or basket in basketball, because they are directly related to players', referees', commentators', and audiences' actions in sports. Particularly, referees' whistling presents some certain judgments and instructions when the game is going on. These actions can be heuristically mapped to some interesting events according to the rules of the specific sports.

- Sleit, A., et al. (2010) The study is elaborated on the problem that the researcher is trying to solve which is most closely related to Azzam sleit work on Text-Based Indexing E-Learning System, they introduced a video search system called Video PowerSearcher which provides the capabilities of searching inside video lectures using text-based search queries. The dataset that be used in their approach is from common craft web site[6]. This system separates audio from video contents and operates a set of first activities to find indexing features for the video which can be used at a later stage for search operations. This system differs from other systems in simplicity, speed, and accuracy but research base is narrow. Figure 16 below shows the structure of Sleit system.

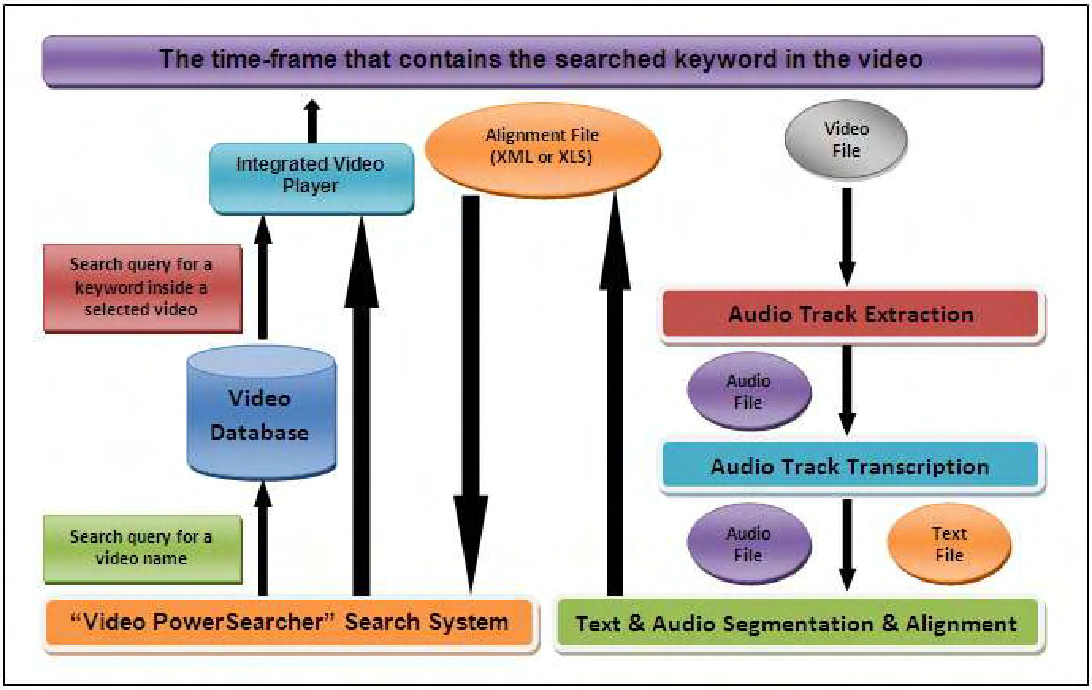


Figure 16. The flow chart of the "Video PowerSearcher" search system

6 http://www.commoncraft.com/videolist, 2014

- Welch, M., et al. (2010) Produced a method for Generating Advertising Keywords from Video Content. They have used text available from video content to get high quality keywords suitable for matching advertisements.
- Farhadi, B., & Ghaznavi-Ghoushchi, M. (2013) propose a new way for semantic indexing subtitled YouTube video content through extracting the main portions from the captions with web natural language processors. They applied SY-VSE (Subtitled YouTube Video Search Engine) as an efficient framework to cruising on the subtitled YouTube videos resident in the Linked Open Data (LOD) cloud and proposed Unifier Module of Natural Language Processing (NLP) Tools Results (UM-NLPTR) for extracting main portions of the 10 NLP web tools from subtitles associated to YouTube videos in order to generate media fragments annotated with resources from the LOD cloud, and they proposed a Unifier Module of Popular API's Results (UM-P AR) containing the seven favorite web APIs to boost results of Named Entities (NE) obtained from UM-NLPTR. They Proposed a web application that called SYVSE by utilizing data model based on NLP web tools, web APIs, ontology and RDF.
- Liu, J., et al. (2013) they focused on Near-duplicate video retrieval. They turned to detect near-duplicate video in a retrieval process on the web as a way to prevent copyright violation. Low-level features, compact signatures (or summaries are the core of their system as shown in figure 17.

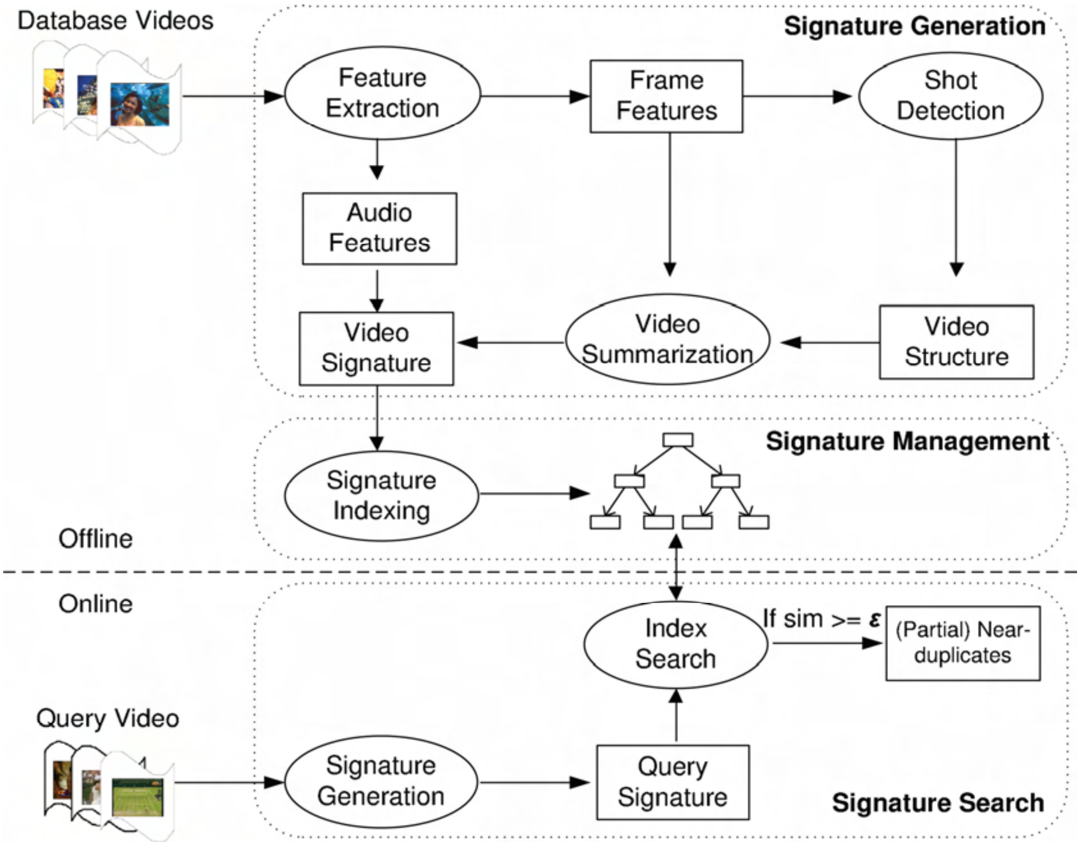


Figure 17. A general framework of NDVR

- Memar, S., et al. (2013) focused on An integrated semantic-based approach in concept based video retrieval. They focused on query based on semantics (example; A female animal can be a tigress, ewe, hen, doe, mother, daughter or sister…) to enhance the retrieval effectiveness in concept-based video retrieval higher level concepts, representation of videos.

- Thepade, S., & Tonge, A. (2014) proposed a method to detection of near duplicates in content based video retrieval to help in reducing the time taken for video processing in large video signals., they work only on visual data, the audio information part is ignored. In their approach they have broken the video into shots, scenes, frames and then key frames which are considered as the base. In their proposed method, they extract the key frames (frame which carry vital information) from an input video structure and only important key frames extracted are evaluated to describe the content of video in the later stages of content based video retrieval. Then they calculated and stored the consecutive difference between 1^{st} and 2^{nd} frame as a difference of first frame (dif_1), and repeated that with 3^{rd} and 4^{th} frame to get (dif_2) and so on for each of the remaining frames. After that they compared the threshold value with each contiguous (dif_n), if it is above threshold the respective frame is considered as key frame and written as output frame. Once Key frames are extracted they determined and eliminated near duplicate copies. They implemented their program by Matlab, used the values of mean, standard deviation and threshold are the base for selection of key frames in their approach.

CHAPTER THREE

The Proposed Technique

3.1. Overview

Using the video in education, is one of the most important means of interaction both direct and indirect, and includes forms of fixed such as slides, shapes animation such as films and videos, in addition to the confusion produced in real time which combines with audio conferencing via video used in one direction or two directions with accompanying sound.

In light of the widespread expansion of knowledge and information as a result of the scientific technological and information revolution (which approached the parties of the globe and spread education), development of institutions, diversity of goals and inclusion of all citizens in the community, (their young and seniors, men and women), as a result of all this, become an urgent necessity to deliver knowledge, science and skill to applicants as soon as in less effort and lower cost, from this standpoint developed plans to achieve better results in the process of learning and teaching. Accordingly, methods and styles are varied to serve purpose according to the required goals. One of these methods is using archived lectures, the importance that through its profound influence on the three key elements of the educational process in which the teacher and the student and the curriculum or article educational.

Archived lectures contain a huge amount of information that will help the user or learner, this irregularly distributed information within the educational material made it difficult to retrieve it in traditional search methods, which depends mainly on hearing all of educational material to reach the required information. As a result, loss of time, increase of effort and probably skip the required information by mistake.

From this background this study stems its to importance helping in retrieving the properly information, retrieve all information that related to research topic, and facilitate the search ways and make information available to everyone (for expert, beginner, learner and illiterate). Using search inside archived video lectures to retrieve the required will increase the effectiveness of learning process among students.

This research facilitating the retrieval of information from multimedia and increasing the chances of acquiring knowledge from these videos archived. It contribute in a quantum leap in the path of self-learning, and gives inspiration to build a good search engine exceeds the existing search engines such as Youtube search engine.

3.2. Methodology of Proposed Solution

This study based on Sleit study and depend on several interrelated developed steps to solve the problem; these are:

Step 1: Conduct a comprehensive literature survey.

Step 2: Adopting an existing technique for extracting audio from video streams, and a mechanism for transforming audio into text by using tools for extracting and transforming.

Step 3: Adopting a mechanism for transforming voice query (query based on voice) into text by using tools for transforming.

Step 4: Extracting related keywords (keyword, Synonyms) from transformed audio using specific built-in synonyms dictionary.

Step 5: Introducing a mechanism for mapping "Multi-Keyword" into video segments (between the source video and the chosen keywords) with adopting an heuristic search technique for finding " Appropriate Nearby Keyword" into video segments.

Step 6: Applying the mapping mechanism on selected e-learning video materials.

3.3. Components of the Proposed System

The proposed system consists of three main plug parts represented as following:

1- Data Preprocessing: This part show the main required steps for preprocessing raw data and make it useful.

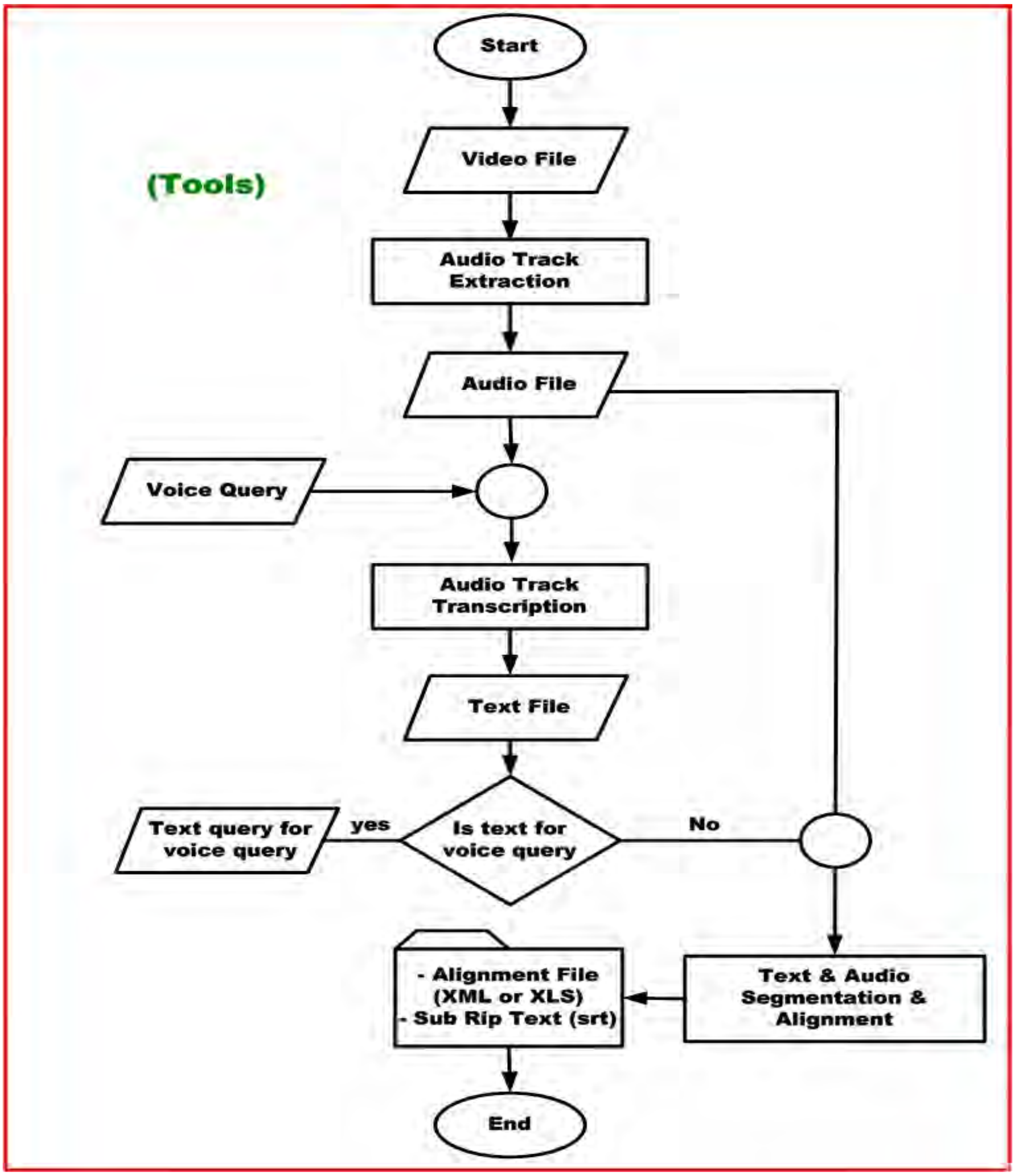


Figure 18. The flow chart of Data Preprocessing

Pseudo Code for Audio-Track-Extraction
input videos
x_1 = count(videos)
set $z_1[x_1]$
$z_1[x_1] = \{videos\}$
for each x_1 in z_1 do
r_1 = audio track extraction
end do
output r_1
Pseudo Code for Audio-Track-Transcription
input r_1 from previous procedure
x_2 = count(r_1)
set $z_2[x_2]$
$z_2[x_2] = \{r_1\}$
for each x_2 in z_2 do
r_2 = audio track transcription // (text result)
end do
output r_2
Pseudo Code for Text/Audio Segmentation & Alignment
Segmentation
input r_1 and r_2 from previous procedures
x_3 = count(r_1)
set $z_3[x_3]$
$z_3[x_3] = \{r_1\}$
for each x_3 in z_3 do
r_3 = number of audio segmentation slices result // depending on silence
end do
set $z_4[x_3]$
$z_4[x_3] = \{r_2\}$
for each x_3 in z_4 do
r_4 = number of text segmentation slices result // depending on number of r_3
end do
Alignment
for each x_3 in (r_3 & r_4) do
r_5 = Text/Audio Alignment result
end do
output r_5

Algorithm 1. Data Preprocessing (Three Steps Sequentially).

2- Voice Recognition: This part required speech recognition to produce a text from voice.

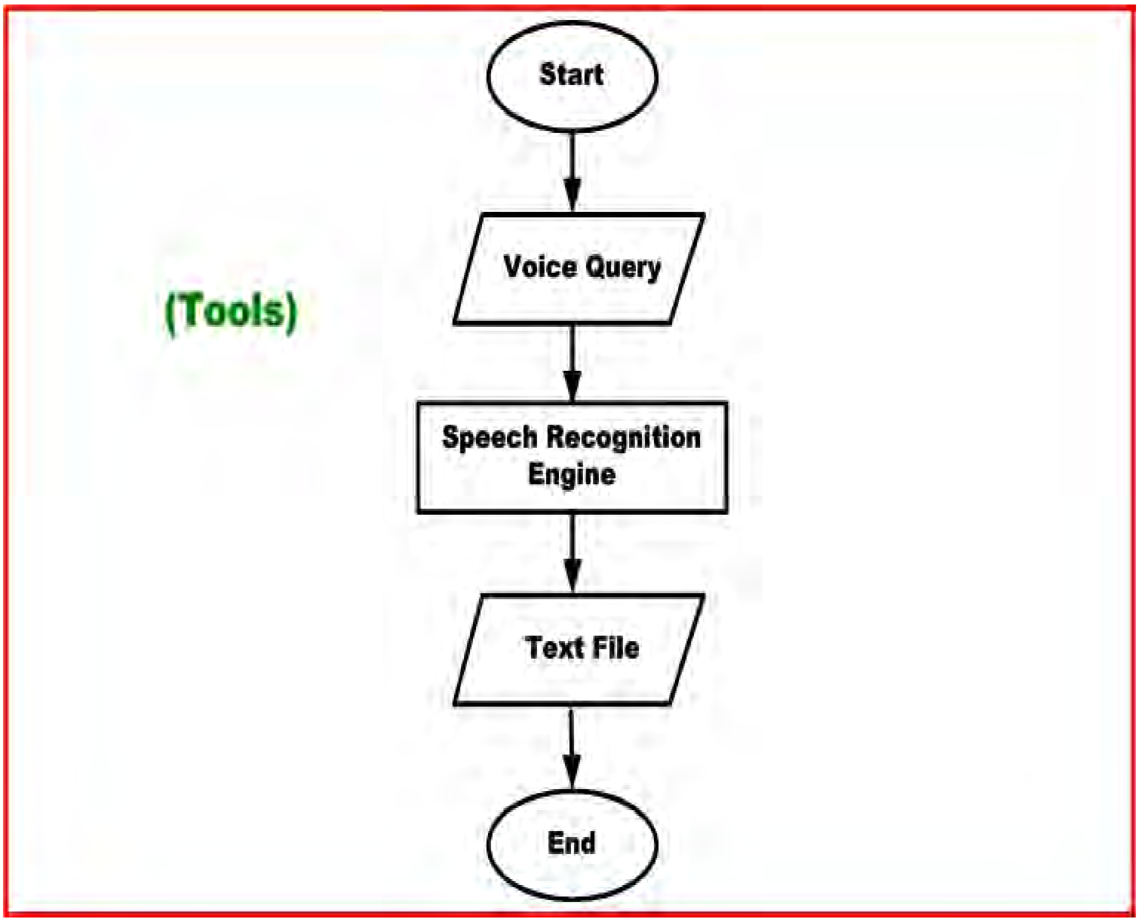


Figure 19. The flow chart of Voice Recognition

Pseudo Code for Voice Recognition

Voice Query for Video(s) Search
let VQ is voice query for video(s) searching
let VQF is voice query for frame(s) searching inside video
let SG is synonyms grammar
let DG is dictation grammar
let DB is videos database
input VQ
if word/words exist in SG // exist in synonyms dictionary
do
s_1 = speech recognition result // according to synonyms grammar
print s_1
end do
else

```
if word/words exist in DG
do
s2 = speech recognition result // according to dictation grammar
print s2
end do
if s1 or s2 exist DB
upload the video(s) in data grid view1
else
null
Voice Query for Frame(s) Search Inside Video
input  VQF
if word/words exist in SG // exist in synonyms dictionary
do
s3 = speech recognition result // according to synonyms grammar
print s3
end do
else
if word/words exist in DG
do
s4 = speech recognition result // according to dictation grammar
print s4
end do
if s3 or s4 exist inside video
upload the video frame(s) in data grid view2
else
null
output s1 and s2 and s3 and s4
```

Algorithm 2. Voice Recognition.

3- The Dictionary: Includes all synonyms words (Approximate 2.5 million words with same meaning exist in one group), and so on for the rest of the words were classified based on the meaning of the word and under "Is a…" relationship, such as home, house, hotel, and hostel placed under same group and same relationship.

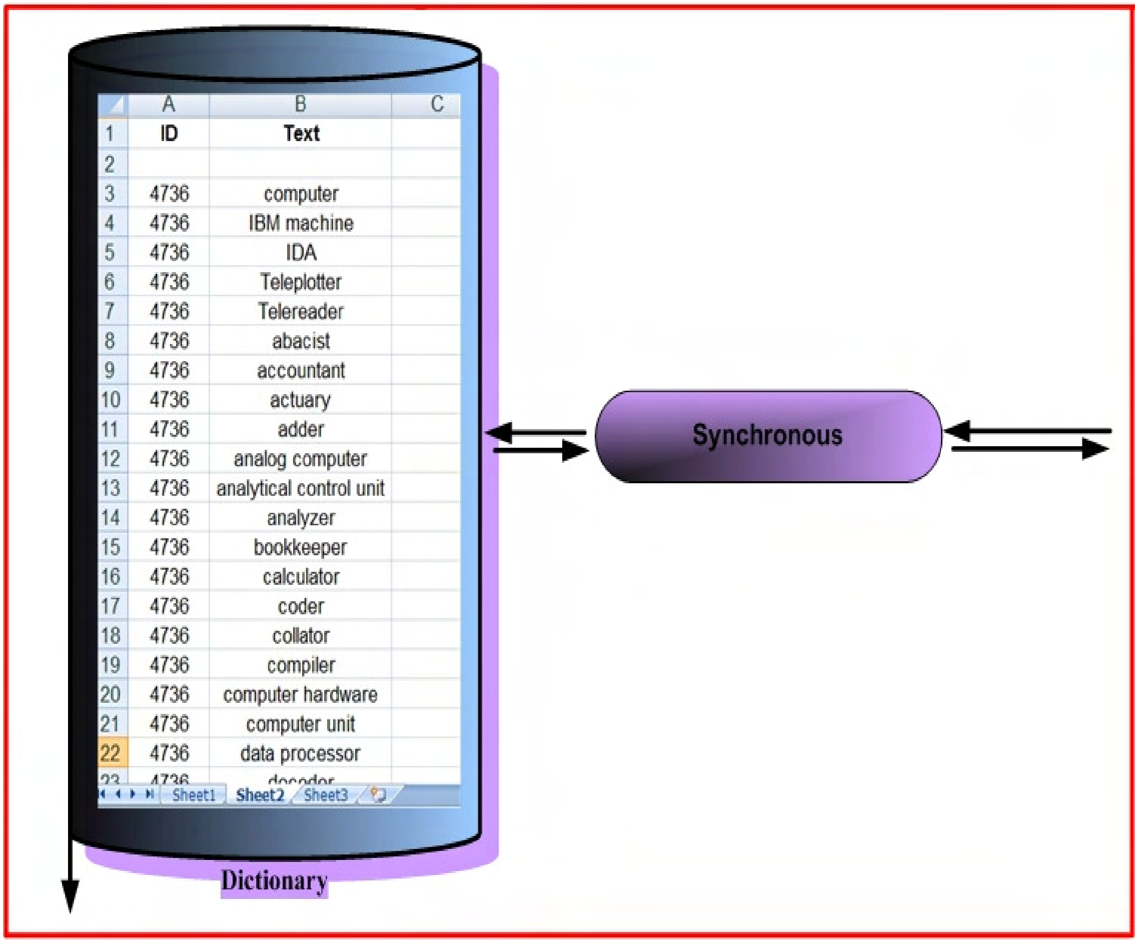


Figure 20. The diagram of Built-in Synonyms Dictionary

Pseudo Code for Built-In Dictionary

ID = 1
for each main word do
d_1 = count(word & their synonyms)
Text[d_1] = {word with their synonyms}
insert into XML file ID and Text[d_1]
ID = ID + 1
end do
output XML

Algorithm 3. Built-In Dictionary.

CHAPTER FOUR

Implementation of Video Multi-Searcher System

4.1. Overview

This chapter will clarify and discuss all the parts and phase of the proposed prototype system, that is capable of searching inside lecture videos, the search capabilities are based on handling audio content of the processed video, which is the speech contained in the video, its search capabilities are based on handling audio content of the processed video, which is the speech contained in the video by proposed system . The phase include:

- Preprocessing phase for videos and make it searchable.
- Phase of building and embed dictionary.
- Searching phase inside videos.

The preprocessing phase for videos is considered indispensable key phase, where basic steps occur to prepare video to become a searchable. These steps include:

- Step 1: Extracting audio track from the video file and make some conversions and improvements.
- Step 2: Transcribing audio track into text.
- Step 3: Segmenting and aligning text to the corresponding audio track (Indexing).

These steps vary between partially user-supervised process, exactly like step 1, and considered fully automatic purely process like step 2, and fully user-supervised process like step 3. But in some cases that will be explained later if several factors are available it can hold above three steps in a single step. Next we will discuss in detail all these steps in addition to proposed system. Figure 21 shows the diagram of the proposed system including data preprocessing.

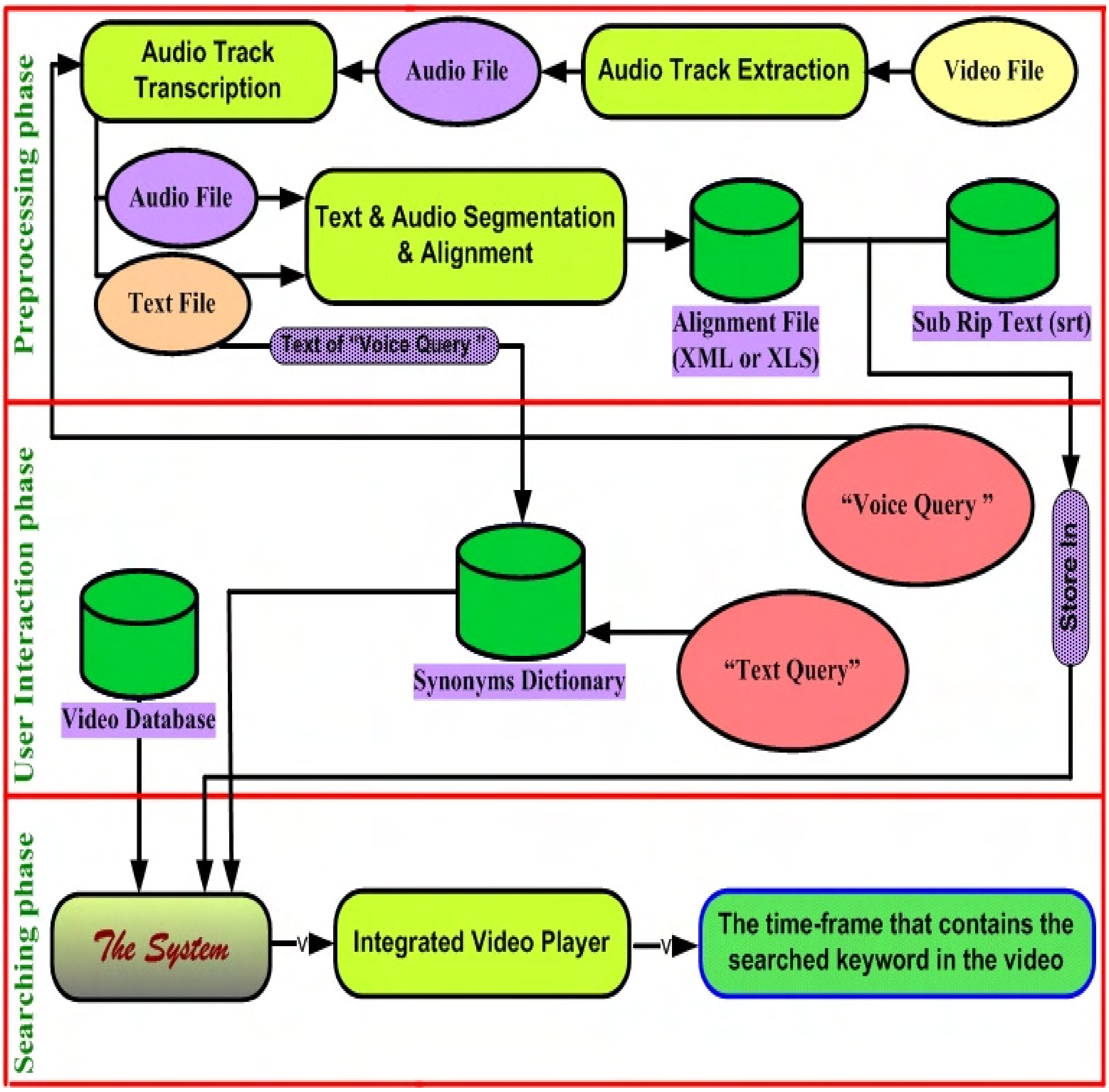


Figure 21. The diagram of the "multi-search" system

The researcher has adopted number of tools to carry out the preprocessing phase; those tools have been chosen carefully to maintain both speed of processing and accuracy of the output data of the three preprocessing steps, none of those tools is system specific.

4.2. Data Preprocessing

At this stage the method that will follow; whether semi-automatic or traditional, is determined by several factors, including the degree of videos confidentiality, and the availability of high-speed of Internet. If videos content includes some sensitive information that must be kept secret or if internet not available, the traditional method is the best, otherwise the semi-automatic method is the best to be followed.

4.2.1. Step 1: Audio Track Extraction

The process of extracting audio from video is a primitive step, however, the proposed system mainly deals with the audio content of video clips and not with motion pictures in the video clip itself. But the quality of the extracted audio content is a critical issue here, because this quality intensively affects the accuracy of the transcription process output (step 2). This system depends on the specific audio file format with a specific audio sample rate, Waveform Audio File Format (.WAV) is the required audio file format to be extracted from the video file in this system. Wave is a Microsoft® and IBM® audio file format standard for storing an audio bit stream on computer. It is considered as the main format used on Windows® systems for raw, high quality sound, lossless and typically uncompressed audio data, but with somewhat big files. Other loosy formats exploit general human hearing to reduce file size, that was the only reason for it to be used, thus causing quality loss. So this audio format is very suitable for processing (Transcription). Sample rate or sample frequency must be 22050 Hz, this value is determined by the ASR engine that transcribes the audio file in the next step (Step 2).

There's a plenty of tools that can be used to extract audio track from video file, in this system we used two tools; the audio track is extracted using Format Factory Version 3.3.5.0® it is a free multifunctional media converter.

Then reshape and other enhancements are applied using Sound Forge Pro 11®, a major reason why we use Sony Sound Forge pro 11 software for audio editing is the excellent set of noise reduction tools. Whether you're dealing with hiss, hum, computer fan noise, or any other constant background sound, Sound Forge software can often reduce it and sometimes, eliminate it. It's advanced audio waveform editor, power, stability, exceptional workflow, and the best way to get from raw audio to a finished master. Expertly record, edit, and analyze audio, produce samples and music loops, digitize and restore old recordings, design sound for multimedia and video, and master replication-ready audio CDs. The following figure depicts a Waveform graph of a Wave audio file, the Y axis represents the amplitude of the audio signal, and the X axis represents time.

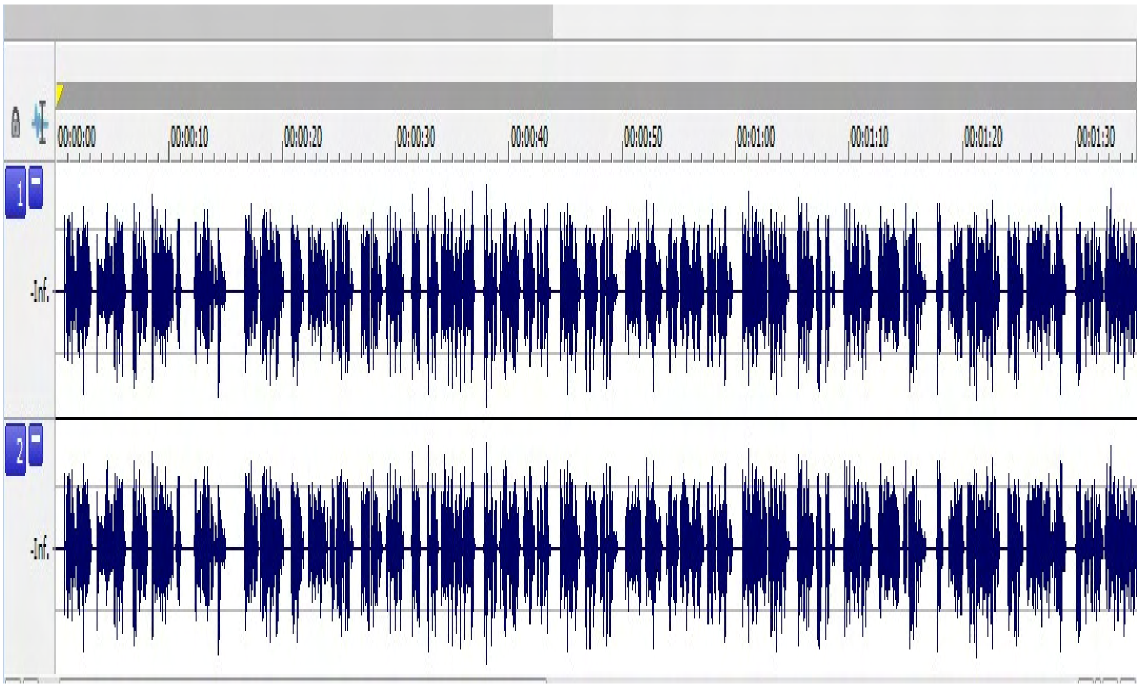


Figure 22. A Waveform graph of a Wave audio file

4.2.2. Step 2: Audio Track Transcription

Audio track transcription which is based on ASR technology that converts spoken words to text is basic and second Step.

The process of speech recognition can be divided concisely into the following consecutive steps (Gruhn, Minker & Nakamura, 2011):

- Pre-Processing (which includes speech/non-speech segmentation).
- Feature extraction.
- Decoding, the actual recognition employing an acoustic and language model as well as a dictionary.

- Result Post-Processing.

Anusuya & Katti (2010) stated that there are several approaches and mechanisms that can be applied in ASR process, those approaches include:

- Acoustic Phonetic Approach.
- Pattern Recognition Approach (Template Based approach, Stochastic Approach).
- Artificial Intelligence Approach (Knowledge Based (Rule based) Approach, Artificial Neural Networks).
- Statistical-Based Approach.

Statistical-based approach is the approach that we use in this system it is based on collecting a large corpus of transcribed speech recordings and then training the computer to learn the correspondences (Machine Learning), and at run time, statistical processes are applied to search through the space of all possible solutions, and the statistically most likely one is picked.

There's a plenty of tools that can be used in this step, such as better wave to text®, adobe premiere pro cs6®, ultra wave to text®, wave to text®, dragon dictate for windows®, and other. In this system we used Wave to Text® from Research Lab Inc. which is a powerful audio transcription tool. This tool is supporting many Speech Recognition Engines, such as Microsoft Speech Recognition Engine 4.0 (English), Microsoft Speech Recognizer 8.0 for windows (English), Microsoft English Recognizer v5.1, Microsoft English v6.1 Recognizer, and other which are a part of Speech Application Programming Interface (SAPI). SAPI is an API that is developed by Microsoft to allow using speech recognition and speech synthesis within Windows applications.

There are three types of mistakes a speech recognition system can make (Gruhn, et al., 2011):

1. Substitution: At the position of a unit (word or phoneme), a different unit has been recognized.
2. Deletion: In the recognition result a unit is omitted.
3. Insertion: The result contains additional units that were not spoken.

The most common measures to evaluate the performance of a speech recognition system are correctness, accuracy, speed (measured with the real time factor), and error (Anusuya et al., 2010).

4.2.3. Step III: Text and Audio Segmentation and Alignment

This step is the last step of data preprocessing, it's consists of two main parts, namely:

- Text segmentation
- Audio-to-text alignment.

Text segmentation part implies segmenting the transcription to small units (words, small sentences, or normal sentences), each piece is stamped with a time frame shown the beginning and the end time. The production of this part is considered as an entrance to the second part. Audio-to-text alignment part contains text segments with the corresponding time frames. In the proposed system, both segmentation and alignment are performed in one step.

There's a plenty of tools that can be used in this step, such as WinPitchPro®, ELAN-Linguistic Annotator®, and other tools. In this proposed system, we use ELAN version 4.7.0® (EUDICO Linguistic Annotator) for segmentation and alignment, this tool provides an excellent environment to perform both segmentation and alignment manually in one step, and creating of complex annotations on video and audio resources, an annotation can be a sentence, word or gloss, a comment, translation or a description of any feature observed in the media, the textual content of annotations is always in Unicode and the transcription is stored in an XML format, with ELAN a user can add an unlimited number of annotations to audio and/or video streams.

As mentioned before, accuracy is a key issue in this system, because it directly affects the quality of the system's search results, so this step is performed manually using this tool since manual alignment produces more accurate results than automatic alignment.

Finally, the output of this step, which comprises the final result of all the preprocessing steps - is stored in an alignment file. Alignment file is an "XML" file that contains text segments with the corresponding time frames. This files will be the main entrance of the proposed system, where we will search within the videos content depending on this files. Figure 23 shows a portion of an XML alignment file.

columns	start	end	v
all_tiers	0.01	3.437	Its easy to think of computers is just machinery
all_tiers	3.437	6.405	buttons screens and parts that work together
all_tiers	6.405	11.755	but the magic of computers: come from something that makes them uniqueto you
all_tiers	11.755	15.472	this is computer software in plain English
all_tiers	15.472	20.434	the problem with is that most of us don't speak their language
all_tiers	20.434	22.157	we need a translator
all_tiers	22.157	26.421	something that can understand our needs and put the computer to work for us
all_tiers	26.421	29.042	the translator is called software
all_tiers	29.042	31.096	and it makes computers useful
all_tiers	31.096	32.591	look at it this way
all_tiers	32.591	37.34	like a typewriter a computer without software is just a lifeless machine
all_tiers	37.34	39.213	by adding software
all_tiers	39.213	41.307	the computer becomes more alive
all_tiers	41.307	43.897	easy to use and built for you
all_tiers	43.897	46.499	most computers have two basic kinds
all_tiers	46.499	49.532	the operating system and software programs
all_tiers	49.532	53.774	if you've ever used a computer you've used in operating system
all_tiers	53.774	59.752	from saving files to using a mouse are fixing problems the operating system covers the basics
all_tiers	59.752	64.957	operating systems come with all new computers and do a lot at the same things
all_tiers	64.957	69.049	you seen them con Windows Mac and Lenux

Figure 23. Example of an XML alignment file

Since data preprocessing is an inevitable step to make videos searchable, we knew that there's some critical issues we must be aware of throughout this step. Accuracy/speed is a classical trade-off issue that affects systems like this system, so we have carefully adopted several tools and techniques to sustain both accuracy in results and speed in execution of this system.

4.3. Semi-Automatic Method

As noted previously, this method provides the speed, good accuracy and summarizes the three steps in a single step. It can be adopted and used if the internet connection is available with fairly at high speeds, and in case that the data embedded within the video files are not confidential or private because the internet is not completely secure. In this method we will use youtube web site to make the three steps above in one step using the following steps:

- Upload the selected video clip to youtube web site.
- Manage the uploaded video by choosing Video Manager button.
- Choose Edit button beside the video name.
- Choose Subtitles and Closed Caption (CC) tab.
- Select video language (in this step we choose English as a default).

- Select English (Automatic) tab.
- Finally save the subtitle by choose Actions button (The output for this step will be file with specific extension (*.vtt, *.srt, and *.sbv).
- After this step we will change the type (Extension) of the output file to XML by using a tool called SubtitleEdit version 3.4.2 build 0®, any time after these steps we can use SubtitleEdit tool to manage the output.

If video does not automatic subtitle or captions, it could be due to one or more of the following reasons:

- The video is too long
- The video has poor sound quality or contains speech that Youtube doesn't recognize; changing video format sometimes solve this problem (Special to avi format with changing audio channel to stereo).
- There is a long period of silence at the beginning of the video
- There are multiple speakers whose speech overlaps

The following figure illustrate all tools with some explanation, that were used during work:

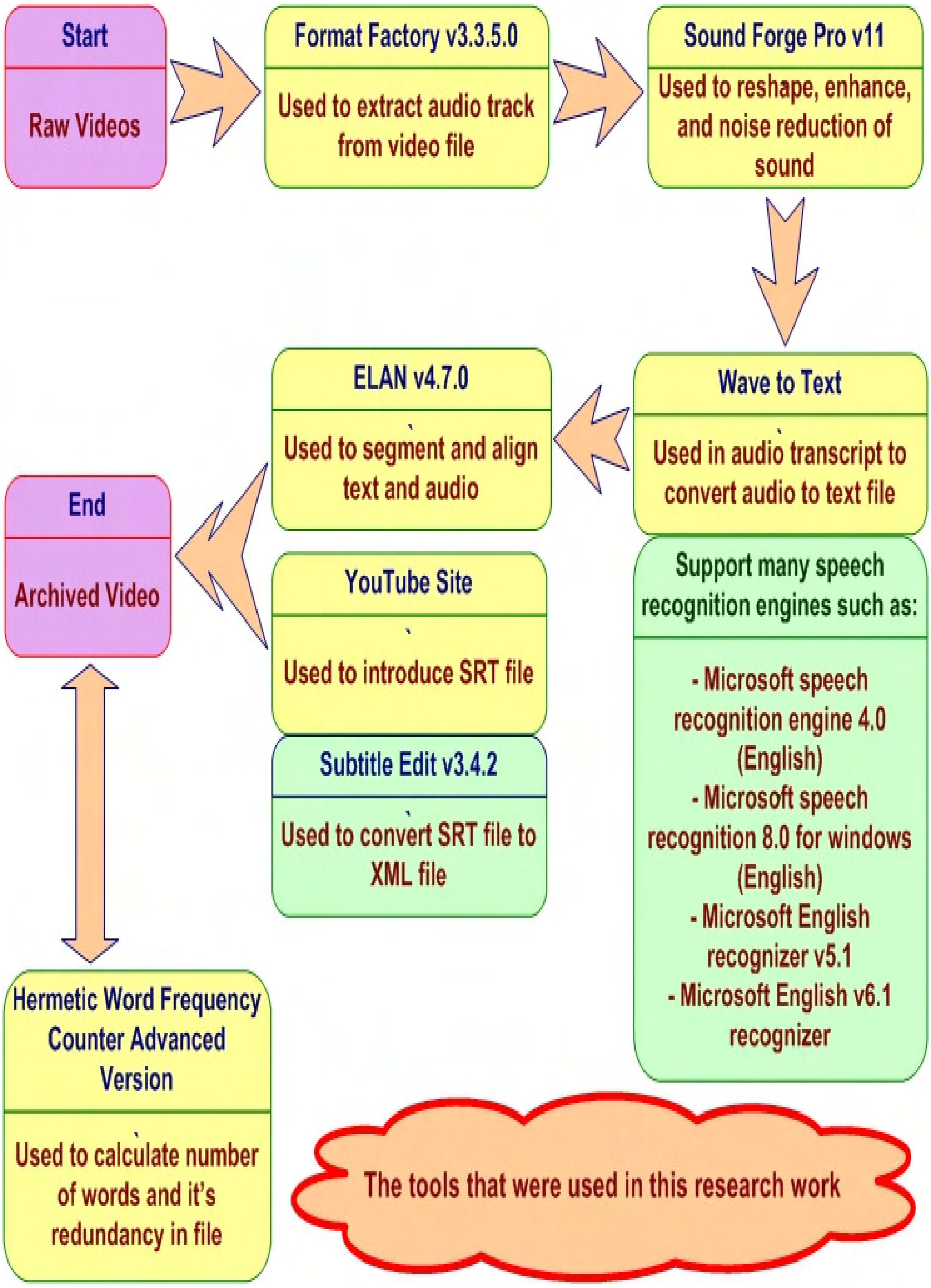


Figure 24. The tools that were used in research work

4.4. Built-in Synonyms Database (Synonyms Dictionary)

Synonym is a word having the same or nearly the same meaning as another in the language, as happy, joyful. A dictionary of synonyms and antonyms (or opposites), is called a thesaurus. A thesaurus is a reference work that lists words grouped together according to similarity of meaning (containing synonyms and sometimes antonyms), in contrast to a dictionary, which provides definitions for words, and generally lists them in alphabetical order. The main purpose of such reference works is to help the user “to find the word, or words, that searched for and give the same meaning. Although including synonyms, a thesaurus should not be taken as a complete list of all the synonyms for a particular word.

In our system we build a private synonyms database to assist in the search process and expanding the research base. By taking advantage of each of Microsoft Word synonyms database (it is a free thesaurus but it does not contain a comprehensive database of synonyms for each existing word) and the database embedded to Mobysaurus Thesaurus Software (includes a database of over 30 thousand root words and phrases and more than 2.5 million synonyms).

The synonyms database built manually depended on one relationship "Is a… relationship" in order to get only the most related results, Thus obtaining positive results in the searching process.

Figure 25 shows a portion of a synonyms database.

ID	Syn_ID	Synonym
424950	4736	computer
424951	4736	IBM machine
424952	4736	IDA
424953	4736	Teleplotter
424954	4736	Telereader
424955	4736	abacist
424956	4736	accountant
424957	4736	actuary
424958	4736	adder
424959	4736	analog computer
424960	4736	analytical control unit
424961	4736	analyzer
424962	4736	bookkeeper
424963	4736	calculator
424964	4736	coder
424965	4736	collator
424966	4736	compiler
424967	4736	computer hardware
424968	4736	computer unit
424969	4736	data processor

Figure 25. Example of Synonyms Database

4.5. The "Video Multi-Searcher" System

The "Multi-search" system comprises the other half of the proposed search system; it provides direct search capabilities through a simple yet practical user interface that allows user to perform five main tasks:

1. Search for a video in the database.
2. Search for a keyword inside a selected video.
3. Search for a query voice inside a selected video.
4. Search synonyms for a keyword inside a selected video.
5. Search for approximate similar keyword inside a selected video.

Figure 26 shows the user interface of Video Multi-Searcher.

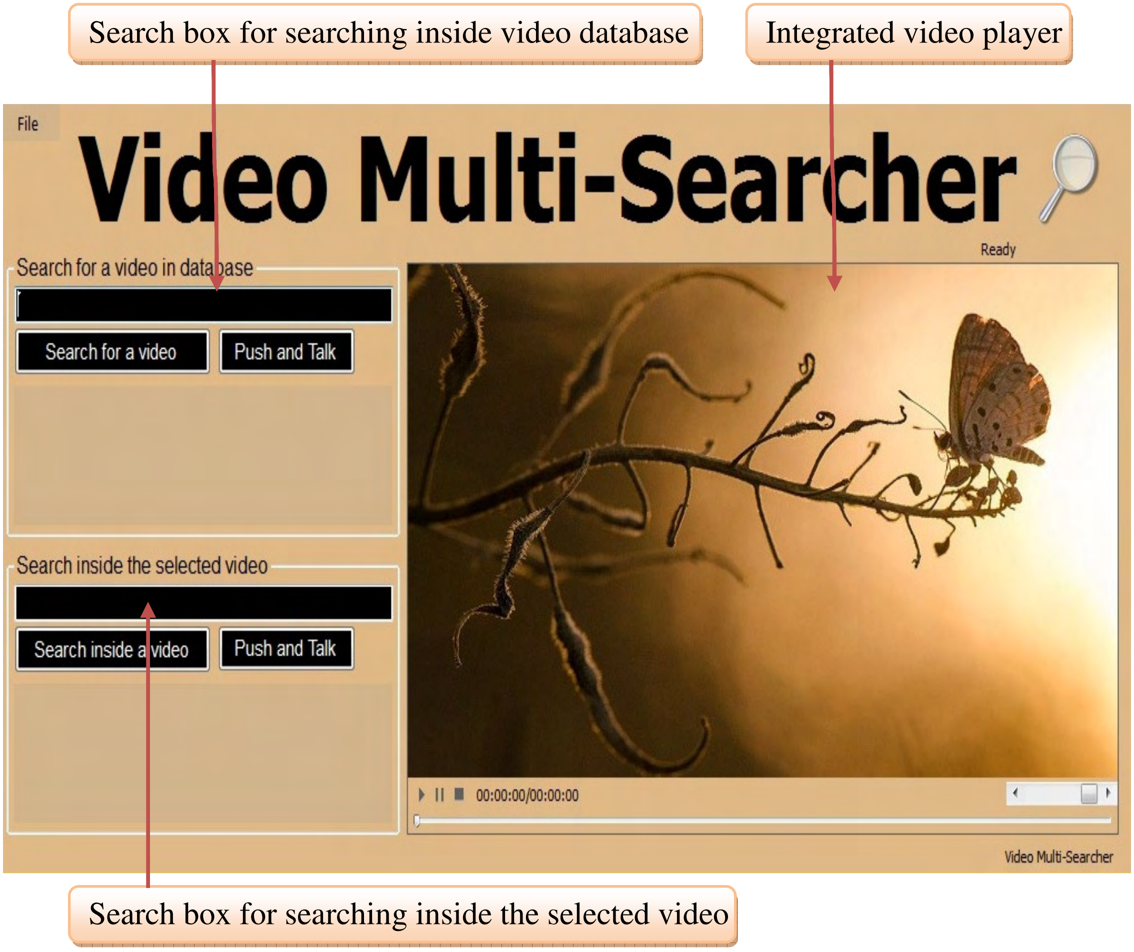


Figure 26. The user interface of Video Multi-Searcher

When using Video Multi-Searcher, the user should at first search for a wanted video through searching for the name of the video, any keyword that's related to the required video, or any part of video name or keyword as shown in figure (27) that shows the search results for a video that contains the word "tutorial" in its name or a video that is categorized as "tutorial".

When searching for a specific video, the system searches for this video in every keyword list in database using the search query. Every video in database has an associated keyword list that contains keywords that are related to the topic of that video (lecture).

In this version of Video Multi-Searcher, those keyword lists are generated manually by the system administrator. Later in this thesis, we introduce some experimental results that can be used in designing an intelligent search engine which will be able to extract keywords from processed videos and infer the topics of those videos (lectures) automatically.

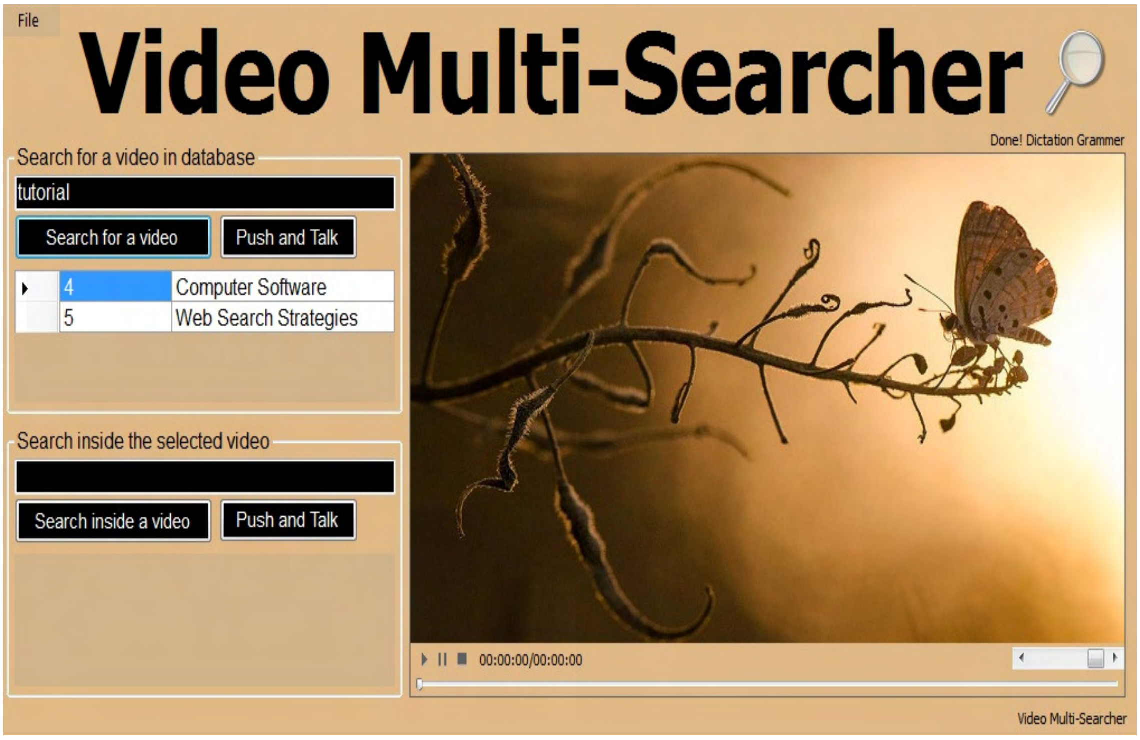


Figure 27. The search results of a search query for the word "tutorial"

In this system the query can be either explicit query by typing a query directly in the box, or by voice where the inquirer speak to system and the system convert the voice to text. Accuracy of the results in the query by voice depends on the accurate pronunciation of the word properly. In this system we adopted two types of grammar in the process of speech recognition there are Dictation Grammar and Synonym Grammar that depend on synonym database which is built-in inside the system. We built synonym grammar to facilitate and increase the accuracy of the speech recognition process. The degree of confidence ranging between 0 an 1, in this system we determined the degree of confidence index equal (0.75) this range is appropriate to give the largest possible accuracy. The researcher experimenting with several values such as 0.50, 0.90, 0.75, and other, the value 0.50 give inaccurate results (scribbles), the value 0.90 give more accurate but it need form speaker to pronounce the word correctly in accurate manner so the system will not be recognize the speech if it does not pronounce correctly, so the value 0.75 will give moderate results between 0.50 and 0.90.

After selecting a video from search results, the selected video will be uploaded into integrated video player, and then the system user can search inside this video for any keyword either by voice or writing. The results of this search will be time frames that match all the segments of the video that contains searched keyword in addition to synonyms of keyword.

Even the sentences that containing the keyword, the system can give all words or sentences that contain the search word. When clicking on one of those time frames, the video player will automatically move the slider to the corresponding segment of the video.

The following figure (28) shows the suggested words for the sub-word “Soft” inside the video “Computer Software”.

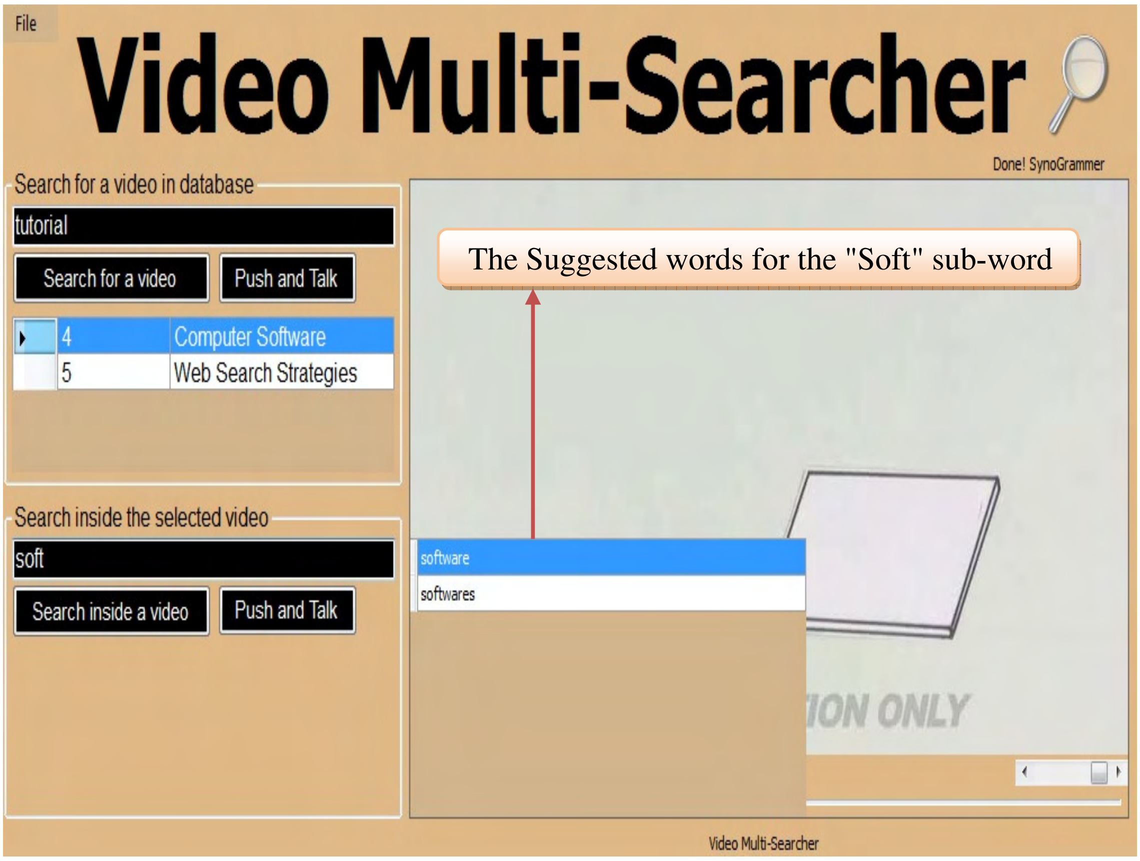


Figure 28. The suggested words for the “Soft” sub-word inside the video

The list of suggested results words will give the user a hint and help him to choice proper keyword. If the user want the sentences that contain the sub-word and the sentences that contain the synonyms word for the sub-word the system will provide that, as shown in figure (29 & 30) below:

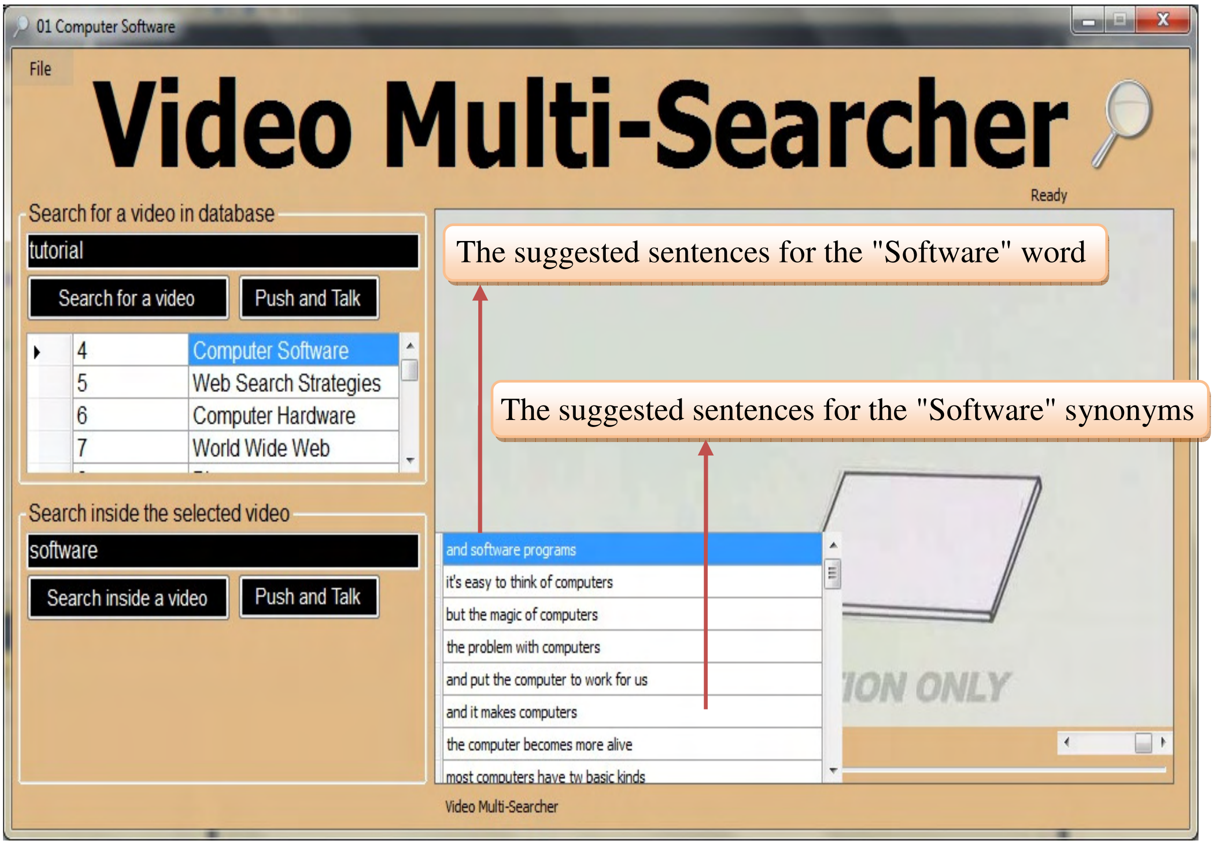


Figure 29. The suggested sentences for the word "Software" and their synonyms inside the video "Computer Software"

The following figure (30) shows the results of a search query for the word "Software" inside the video "Computer Software". The system will give all the matched keyword results in addition to all related synonyms result.

Since searching a file is much faster than searching a database, video file pathname, alignment file pathname, and video's keyword list are all stored in an integrated database which provides simple access and quick response.

The dataset that be used in this thesis is the same dataset that used by the Sleit researcher from common craft web site. The user can be writing the keyword inside searching box that searching inside the selected video by either voice or direct writing.

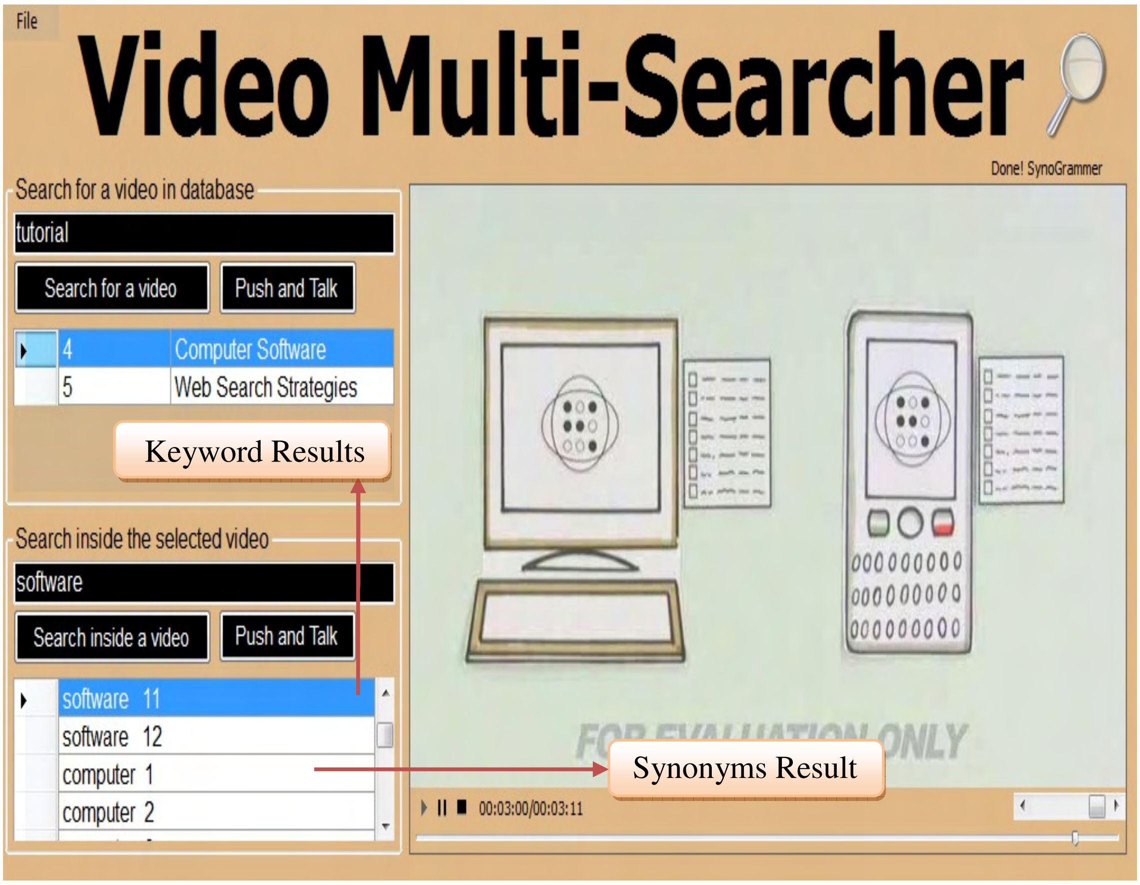


Figure 30. Results of a search query for the word “Software” and it's synonyms inside the video “Computer Software”

The figure below shows the flowchart of user interaction with the system from the beginning to the end, and shows the steps that performed by the user on the system:

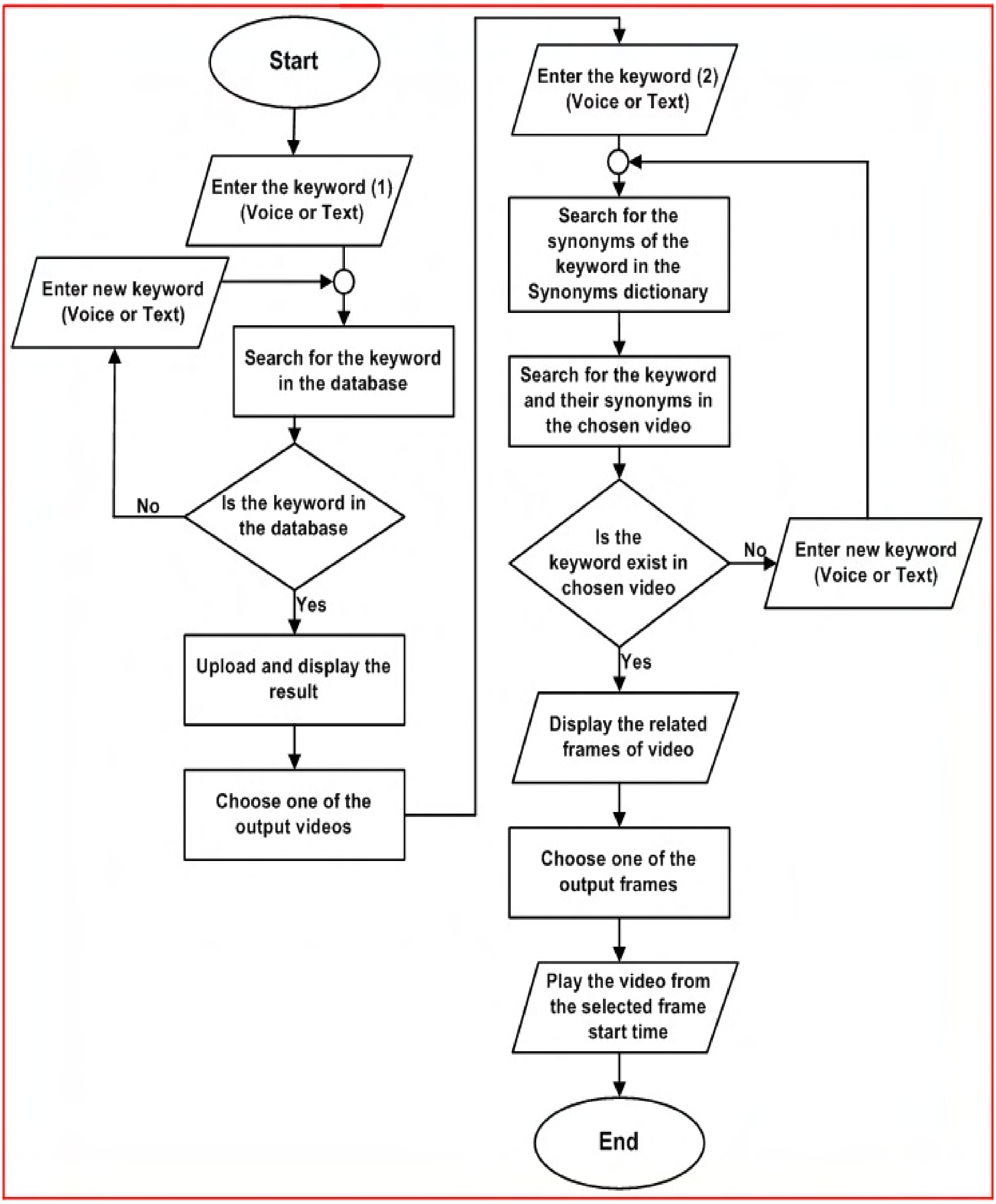


Figure 31. The flowchart of user interaction with the system

4.6. Summary

Video Multi-Searcher is a prototype system that is mainly intended for searching inside video archives of lectures. Data preprocessing is a fundamental step to prepare data to be searched by Video Multi-Searcher, this step includes audio extraction, audio transcription, and audio/text segmentation and alignment.

Poorly recorded video produces a transcription with a high error rate, so Video Multi-Searcher assumes that input videos were recorded in good-quality conditions using good quality devices, these conditions apply to the majority of newly video recorded lectures.

The accuracy results of the query by voice depends on the accuracy of the pronunciation of the word properly, this means that the process of speech recognition depends on the extent of mastering the speaker to speak. To avoid problems with speech recognition we built and used primary grammar in speech recognition process called SynoGrammar depend on synonyms database in order to improve results, and secondary grammar in speech recognition process called DictationGrammar that submitted by existing programming languages in order to speech recognize in the traditional way.

In Video Multi-Searcher, user administration is required to perform two main tasks:

- Performing the user-administrated parts of data preprocessing.
- Maintaining video database.

CHAPTER FIVE

Experimental Results

5.1. Experiments' Methodology

Some statistical experiments were conducted on some archived video lectures using the proposed system, the main purpose of these experiments concentrated on two axes, the first axis: is to find number of occurrences some selected keywords and their synonyms in these films. The second axis: is to count the number of times that the system identified the voice query correctly (Correctly speech recognition – calculated under precision measure). In the first axis experiments were divided into two main parts; first part: calculating the ratio between number of occurrences of a given keyword and their synonyms in the selected video relative to total number of words in these video, the first part can help in chosen the main keywords of a given video depending on these results. The second part: calculating the ratio between the number of occurrences of a given keyword and their synonyms and the overall duration of its corresponding video. The second part can help in designing an intelligent search engine that can infer the main topic of a given video depending on these results.

The following formulas illustrate all of above:

$$\text{The result} = \frac{\text{Number of occurrences}}{\text{Total number of words in video}} * 100\%$$

$$\text{The result} = \frac{\text{Number of occurrences}}{\text{The duration of video}} * 100\%$$

$$\text{Precision} = \frac{\text{Number of true positives (Speech recognition correctly)}}{\text{Number of true psitives} + \text{false positives}} * 100\%$$

Experiments were performed on (16) different videos, (6) different words in text search query that have been tested for each video which is equivalent to (96) words in total, and (15) different word in voice search query that have been tested. The total number of words that have been used in testing are (111) different words, and the total time of videos duration was (2826) seconds, we think it will be enough test set and cover most of topics. The following table (2) shows the numbers of occurrences of the most frequent words mentioned in some videos which were selected randomly from video lists.

Table 2: The numbers of occurrences of the most frequent words mentioned in some videos

#	Video Name	Keyword	Number of Occurrences
1	**Computer Viruses and Threats**	Computer(s)	19
		Virus(s)	11
		Software	6
2	**Plagiarism**	Idea(s)	11
		Credit	5
		Plagiarism	4
3	**Programming Languages**	Computer(s)	16
		Language	8
		Programming	8

Video 1 which discusses "Computer Viruses and Threats" has 521 words[7] (The total number of words calculated using free existing tool called Hermetic Word Frequency Counter Advanced Version, including the words of one, two, or more character length) and (193 sec) duration. As seen in table 2, in first part by calculating the ratio between the number of occurrences of the most frequent words and the total number of words in this video, and by calculating the ratio between the number of occurrences of the most frequent words with their synonyms and the total number of

7 http://www.hermetic.ch/wfca/wfca.htm, 2014

words in this video, the researcher found improvement ratio as is presented in the following table.

Table 3: The ratios of occurrences of the most frequent words and their synonyms mentioned in "Computer Viruses and Threats" video

#	Keyword	Ratio between the keyword and the total words in the video	Ratio between (the keyword and their synonyms) and the total words in the video	Improvement ratio (The difference between the two ratios)
1	Computer(s)	3.65% of words	5.37% of words	1.72%
2	Software	1.15% of words	5.37% of words	4.22%
3	Virus(s)	2.11% of words	2.88% of words	0.77%

In second part by calculating the ratio between the number of occurrences of the most frequent words and the duration of the video (in seconds), and by calculating the ratio between the number of occurrences of the most frequent words with their synonyms and the duration of the video (in seconds), the researcher found that these words were mentioned as presented in table 4.

Table 4: The ratios of occurrences of the most frequent words and their synonyms mentioned in "Computer Viruses and Threats" video for the total time

#	Keyword	Ratio between the keyword and the duration of the video (in seconds)	Ratio between (the keyword and their synonyms) and duration of the video (in seconds)	Improvement ratio (The difference between the two ratios)
1	Computer(s)	9.84% of video	14.51% of video	4.67%
2	Software	3.11% of video	14.51% of video	11.40%
3	Virus(s)	5.70% of video	7.77% of video	2.07%
This video falls under "Software" topic				

In video 2 which discusses "Plagiarism" has 425 words (Including the words of one, two, or more character length) and (152 sec) duration. As seen in table 5, in first part by calculating the ratio between the number of occurrences of the most frequent words and the total number of words in this video, and by calculating the ratio between the number of occurrences of the most frequent words and their synonyms and the total number of words in this video, the researcher found improvement ratio as shown in table 5.

Table 5: The ratios of occurrences of the most frequent words and their synonyms mentioned in "Plagiarism" video

#	Keyword	Ratio between the keyword and the total words in the video	Ratio between (the keyword and their synonyms) and the total words in the video	Improvement ratio (The difference between the two ratios)
1	Idea(s)	2.59% of words	4.24% of words	1.65%
2	Credit	1.18% of words	5.88% of words	4.70%
3	Plagiarism	0.94% of words	1.18% of words	0.24%

In second part by calculating the ratio between the number of occurrences of the most frequent words and the duration of the video (in seconds), and by calculating the ratio between the number of occurrences of the most frequent words and their synonyms and the duration of the video (in seconds), the researcher found that this words were mentioned as shown in table 6.

Table 6: The ratios of occurrences of the most frequent words and their synonyms mentioned in "Plagiarism" video for the total time

#	Keyword	Ratio between the keyword and the duration of the video (in seconds)	Ratio between (the keyword and their synonyms) and duration of the video (in seconds)	Improvement ratio (The difference between the two ratios)
1	Idea(s)	7.24% of video	11.84% of video	4.60%
2	Credit	3.29% of video	16.45% of video	13.16%
3	Plagiarism	2.63% of video	3.29% of video	0.66%
This video falls under "Credit" topic				

In third video the researcher took occurrences of the most frequent two-word phrases mentioned in video shown in table 7.

Table 7: The numbers of occurrences of the most frequent two-word phrases mentioned in "Programming Languages" video

#	Video Name	Two-Word Phrases	Number of Occurrences
1	**Programming Languages**	Programming Language	3
		Machine Language	2
		Computer Device	1

The total of different phrases is (282) and the total duration of video is (161 sec). The ratio of occurrences of the most frequent two-word phrases mentioned in the video according to the total of different phrases and the total duration of video is shown in table 8.

Table 8: The ratios of occurrences of the most frequent two-word phrases mentioned in "Programming Languages" video

#	Two-Word Phrases	Ratio between (the keyword and their synonyms) and the total of different phrases in the video	Ratio between (the keyword and their synonyms) and duration of the video (in seconds)
1	Programming Language	1.06% of words	1.86% of video
2	Machine Language	0.71% of words	1.24% of video
3	Computer Device	0.35% of words	0.62% of video

As seen in tables (3 to 8), using keywords with their synonyms will improve the searching process, and give a hint about the topic of video.

In the second axes the researcher found ratio of speech recognizing for voice query at the beginning three videos have been chosen as a checking and testing set, from these set the researcher choose five keywords for experimenting from every video, every keyword has been tested ten times, at the end ratio calculated in proposed system (Identify words correctly, calculated by counting the number of times the system recognized the voice query correctly relative to the total number of times pronunciation the word).

The results were positive, table (9) shows the results of speech recognition process (Voice Query). The result in voice query depended on several factors, including accuracy pronounce the word correctly, clarity of the speaker voice, lack noise and interference.

The precision measure used in speech recognition process for voice query, it calculated under this formula:

$$\text{precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false positives}}$$

precision or positive predictive value is defined as the proportion of the true positives (speech recognition correctly) against all the positive results (both true positives and false positives of speech recognition)

An accuracy of 100% means that the measured values are exactly the same as the given values.

Table 9: The ratios of speech recognition correctly for some keywords mentioned in some videos

#	Video Name	Keyword	The ratio of speech recognition correctly ((# of speech recognition correctly / total # of trying)* 100%)
1	Computer Software	Software	70%
		Machine	90%
		English	70%
		Useful	60%
		Speak	100%
2	Computer Viruses and Threats	Virus	80%
		Computer	30%
		Damage	50%
		Disease	70%
		Example	90%
3	Search Engine Optimization (SEO)	Engine	80%
		Exact	70%
		Imagine	100%
		Day	20%
		Big	60%
Arithmetic average			69.30%

As shown in the above table the system recognized many voice queries differently depending on some factors that have been noted previously, these results helped and inspired the creation of an intelligent voice search engine (search by voice), depending on voice instead of typing the query.

5.2. Comparison

In this study, the researcher comparing his results with the results of the researcher Sleit because this study is based upon the researcher Sleit study, but it measure under just one axis depend on the ratio between the key word and the duration of the video, the results shown in table below illustrate that:

Table 10: The comparison of result between the researcher study and Sleit study

#	Keyword	Ratio between the keyword and the duration of the video (in seconds) according Sleit researcher	Ratio between (the keyword and their synonyms) and duration of the video (in seconds) according this study	Improvement ratio (The difference between the two ratios)
1	Computer(s)	9.84% of video	14.51% of video	4.67%
2	Software	3.11% of video	14.51% of video	11.40%
3	Virus(s)	5.70% of video	7.77% of video	2.07%
Computer Viruses and Threats Video				
1	Idea(s)	7.24% of video	11.84% of video	4.60%
2	Credit	3.29% of video	16.45% of video	13.16%
3	Plagiarism	2.63% of video	3.29% of video	0.66%
Plagiarism Video				

As an example from table (10) if searching process done based on software keyword in first video as Sleit study the result will be (3.11%) of video duration, but in proposed system this result will increase to be reached up to (14.51%) of video duration if software keyword took with it synonyms, this means improve in result up to (11.40%), so the proposed system enhance the result, thus increase the benefit of video. As an another example if searching process done based on credit keyword in second video as Sleit study the result will be (3.29%) of video duration, but in proposed system this result will increase to be reached up to (16.45%) of video duration if software keyword took with it synonyms, this means improve in result up to (13.16%). Overall, the proposed system enhance multimedia information acquisition and retrieval process.

5.3. Results

The researcher found in first axes that the results vary from (3% - 9%) if the searching is just by the keywords, but the result will grow to reach from (4% - 16%) if the searching done by the keywords and their synonyms occurrences of the most frequent words and their synonyms mentioned in videos for the total time. These results can be very helpful in designing an intelligent search engine that can infer the main topic of the processed video. Let us take video (2) as an example, the duration of the video is known, and the most frequent words like "Idea" or "Plagiarism" can be found using the default search engine in video Multi-Searcher, so the system can easily calculate the ratio between the numbers of occurrences of those words and the long of the video, and depending on the results of these calculations the system will decide whether to take any of those words as a keyword or not. The results vary from (0.9% to 3.7%) if the searching is just by the keywords, but the result will grow to reach from (2.9% to 5.9%) if the searching is done by the keywords and their synonyms occurrences of the most frequent words and their synonyms mentioned in videos for the total words in the videos. In second axes the results helped and inspired the creation of an intelligent voice search engine (search by voice), depending on voice instead of typing the query.

The system can be trained to neglect the meaningless words like "The" or "In" or "Of" or any other similar words that can't be used as keywords, the system also can be trained to adopt different types of lectures and topics which allow it to handle different (words frequencies/duration) ratios to improve the accuracy of results. These ratios may vary according to the lecture topic, lecturer, or lecture long, the system should be designed to be customized to handle all those lecture properties.

5.4. Contributions

Most of methods for selecting the keywords for a video often rely on user supplied metadata, such as the video title, summary, comments, anchor text from adjacent pages, and so on. These outdated methods designed to perform specific purposes and reflect the need to an audio indexing search system that allows end users to search and navigate this video material based on audio content and system treating individual weaknesses in this methods.

"keyword" search may not return all the relevant search results as if seeking for a home and not return house, or if the item to be searched for within the audio content is a "voice query" and not a specific keyword, from this place "multi-search" query based on voice, and keywords with their synonyms will improved the result of search operations especially in archived E-Lecture, facilitated the retrieval of information from multimedia, increased the chances of acquiring knowledge from archived videos, and contributed in a quantum leap in the path of self-learning.

CHAPTER SIX

Conclusion and Future Work

6.1. Conclusion

Recent advances in speech recognition and natural language processing technologies are making it increasingly possible to allow schools, colleges, and universities to build huge archives of video lectures, and to browse, search, summarize, and extract information from those lectures quickly and practically. This thesis has introduced and described Video Multi-Searcher, which is a partially-supervised prototype search system that is mainly intended for enhance searching inside video archives of lectures. Video Multi-Searcher takes a text-based search query and return the time frame in which the searched keywords and their synonyms were mentioned inside the browsed video, this thesis also has described the underlying technology of this system.

Statistical experiments were carried out in this thesis on (16) different videos with (111) different test words and the results improved varies from (2.7% to 5.9%) of the results obtained by the researcher Sleit. Those results can inspire designing an intelligent search engine that can infer the topic of video lectures and decide the main keyword automatically.

6.2 Future Work

- Adopting different ASR engine that can provide higher level of accuracy special in voice query.
- Training the ASR engine to handle specific speakers (lecturers).
- Integrating data preprocessing steps into the main system.
- Releasing Video Multi-Searcher as a web service through integrating it into a website so it can handle uploaded video lectures automatically.
- Exploiting the experimental results which were introduced in this thesis to design an intelligent search engine that can infer the topic of video lectures automatically.
- Improve and index embedded dictionary in proposed system to increase the speed of searching and retrieving.
- Releasing Arabic Video Multi-Searcher copy that can search in Arabic videos and building built-in Arabic dictionary for that.
- Releasing Video Multi-Searcher as a service application on smart phone devices.

References

- Aladwan, A., & Smedley, J. (2012). *"Implementing e-learning in the Jordanian Higher Education System: Factors affecting impact"*. ***IJEDICT***, *8*(1), 121-135.
doi: http://ijedict.dec.uwi.edu/viewissue.php?id=31

- Anusuya, M., & Katti, S. (2010).*" Speech Recognition by Machine, A Review"*. ***International Journal of Computer Science and Information Security (IJCSIS),*** *6*, 181-205.
doi: http://arxiv.org/abs/1001.2267v1

- Beaufays, F., Sankar, A., Williams, S., & Weintraub, M. (2003). *"Learning Name Pronunciations in Automatic Speech Recognition Systems"*. ***IEEE, 15th International Conference on Tools with Artificial Intelligence***, 233-240.
doi: http://doi.ieeecomputersociety.org/10.1109/TAI.2003.1250196

- Buchsbaum, A., & Giancarlo, R. (1997).*" Algorithmic aspects in speech recognition: an introduction "*. ***Journal of Experimental Algorithmics (JEA)***.
doi: http://dl.acm.org/citation.cfm?doid=264216.264219

- Cao, L. (2010). *"Domain Driven Data Mining: Challenges and Prospects"*. ***IEEE Knowledge and Data Engineering***, *22*(8), 755-769.
doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.83.8505

- Coden, R., & Brown, E. (2001). *"Speech Transcript Analysis for Automatic Search"*. ***hicss, 34th Annual Hawaii International Conference on System sciences***, *4*.
doi: http://doi.ieeecomputersociety.org/10.1109/HICSS.2001.926473

- Colbath, S., Kubala, F., Liu, D., & Srivastava, A. (2000). *"Spoken Documents: Creating Searchable Archives from Continuous Audio"*. ***hicss, 33rd Hawaii International Conference on System Sciences***, *3*, 3005.
doi: http://doi.ieeecomputersociety.org/10.1109/HICSS.2000.926688

- Collis, B., Nikolova, I., & Martcheva, K. (1995). ***"Information technologies in teacher education"***.(Illustrate ed.) Paris: United Nations Educational Scientific And Cultural Organization (Unesco).
doi: http://www.isbns.cc/isbn/9789231030727

- Cooper, P. (2014). *"Data, information, knowledge and wisdom"*. ***Anaesthesia & Intensive Care Medicine*** , *15*(1), 44-45.
doi: http://www.anaesthesiajournal.co.uk/article/S1472-0299(13)00305-6/fulltext

- Demetriadis, S., & Pombortsis, A. (2007). *" E-lectures for flexible learning: a study on their learning efficiency "*. ***Journal of educational technology and society***, 10, 147-157.
doi: http://www.editlib.org/p/75298/

- Dunn, J. (2012). *"***The Importance Of Internet Access In Schools***"*.(On-Line)
available: http://www.edudemic.com/the-importance-of-internet-access-in-schools/

- Elmasri, R., & Navathe, S. B. (2010). ***"Fundamentals of Database Systems"***.(6th ed.). USA: Addison-Wesley Publishing Company.
doi: http://dl.acm.org/citation.cfm?id=1855347

- Erol, B., & Li, Y. (2005). *"An overview of technologies for e-meeting and e-lecture"*. ***Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)***.
doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.83.8505

- Farhadi, B., & Ghaznavi-Ghoushchi, M. (2013). *"Creating a novel semantic video search engine through enrichment textual and temporal*

features of subtitled YouTube media fragments". ***Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference***, 64 - 72.
doi: 10.1109/ICCKE.2013.6682857

- Geoffrey, R., & Mactavish, A. (2007). *"Multimedia"*. In S. Susan, R. Siemens & J. Unsworth (Eds.), ***A Companion to Digital Humanities*** (pp. 108-120): Blackwell Publishing Ltd.
doi: 10.1002/9780470999875.ch10
- Gonzalez, T., Diaz-Herrera, J., & Tucker, A. (2014). *"**Computing Handbook: Computer Science and Software Engineering**"*. (3rd ed.). United Kingdom: Chapman and Hall/CRC.
doi: http://dl.acm.org/citation.cfm?id=2648632
- Gruhn, R. E., Minker, W., & Nakamura, S. (2011). ***"Statistical Pronunciation Modeling for Non-Native Speech Processing (Signals and Communication Technology)"***. London, New York: Springer.
doi: 10.1007/978-3-642-19586-0
- Hajaya, M. (2010). *"**Spoken Language Processing for Video Archives of Lectures**"*. (Unpublished Master Thesis), Jordan University, Amman, Jordan.
- Han, J., & Kamber, M. (2006). *"**Data mining concepts and techniques**"*. (2nd ed.). San Francisco: Morgan Kaufmann Publishers.
doi:http://www.archive.org/details/DataMiningConceptAndTechniques2nd Edition
- Haubold, A., Natsev, A., & Naphade, M. (2006). *"Semantic Multimedia Retrieval using Lexical Query Expansion and Model-Based Reranking"*. ***Multimedia and Expo, IEEE International Conference*** , 1761 - 1764.
doi: 10.1109/ICME.2006.262892

- Hirzallah, N. (2007). *"An Authoring Tool for As-in-Class E-lectures in E-learning Systems"*. ***American Journal of Applied Sciences**, 4*(9), 686-692. doi: http://www.thescipub.com/abstract/10.3844/ajassp.2007.686.692

-Jinjun, W. (2006). *"***Content-Based Sports Video Analysis and Composition***"*. (Unpublished Dissertation Thesis), Nanyang Technological University, Singapore.

- Kawahara, T. (2004). *"Spoken Language Processing for Audio Archives of Lectures and Panel Discussions"*. ***International Conference on Informatics Research for Development of Knowledge Society Infrastructure (ICKS'04)***, 23-30. doi:http://www.computer.org/csdl/proceedings/icks/2004/2150/00/21500023-abs.html

- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2004). *"Multimedia Mining*". ***WSEAS Transactions on Systems**, 3*, 3263-3268. doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.6942

- Kowalski, G. (1997). *"**Information Retrieval Systems: Theory and Implementation**"* (1st ed). USA: Kluwer Academic Publishers Norwell. doi: http://dl.acm.org/citation.cfm?id=523964

- Leavitt, N. (2002). *"Let's Hear It for Audio Mining"*. ***Computer**, 35*(10), 23-25. doi: 10.1109/MC.2002.1039511

- Lee, D., Kim, H.-s., Kim, E. K., Yan, S., Chen, J., & Lee, J. (2008). *"LeeDeo: Web-Crawled Academic Video Search Engine"*. ***IEEE, Tenth IEEE International Symposium on Multimedia***, 497 - 502. doi: 10.1109/ISM.2008.105

- Liu, J., Huang, Z., Cai, H., Shen, H., Ngo, C., & Wang, W. (2013). *"Near-Duplicate Video Retrieval: Current Research and Future Trends"*. ***ACM Computing Surveys (CSUR)**, 45*(4), 44.41-44.23.

doi: 10.1145/2501654.2501658

- Lucas, P., & Gaag, L. (1991). *"Principles of Expert Systems"*. ***Addison-Wesley Longman Publishing Co***.

doi: http://dl.acm.org/citation.cfm?id=132689

- Maji, S., & Bajcsy, R. (2007). *"Fast unsupervised alignment of video and text for indexing/names and faces"*. ***MS '07 Workshop on multimedia information retrieval on The many faces of multimedia semantics***, 57-64.

doi: 10.1145/1290067.1290077

- Memar, S., Affendey, L., Mustapha, N., Doraisamy, S., & Ektefa, M. (2013). *"An integrated semantic-based approach in concept based video retrieval"*. ***Journal. Multimedia Tools and Applications***, *64*(1), 77-95.

doi: 10.1007/s11042-011-0848-4

- Najjar, L. (1996). *"Multimedia information and learning"*. ***Journal of Educational Multimedia and Hypermedia***, *5*, 129-150.

doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.1654

- Naidu, S. (2003). *" **E-Learning: A Guidebook of Principles, Procedures and Practices**"*. New Delhi, India: Commonwealth Educational Media Centre for Asia (CEMCA).

doi: http://dspace.col.org/handle/123456789/138

- Neuro AI, 2013.*"***Speech Recognition** *"*. (On-Line)

available:http://www.learnartificialneuralnetworks.com/speechrecognition.html

- Owaied, H., & Abu-A'ra, M. (2007). *"Functional Model of Human System as Knowledge Based System"*. ***International Conference on Information & Knowledge Engineering, IKE***. 158-164.

doi: http://dblp.uni-trier.de/db/conf/ike/ike2007.html#OwaiedA07

- Rouse, M., 2005.*"***Audio***"*. (On-Line)

available: http://whatis.techtarget.com/definition/audio

- Rouse, M., 2005.*"***Digital Video***"*. (On-Line)
available: http://searchsoa.techtarget.com/definition/digital-video

- Stair, R., Moisiadis, F., Genrich, R., & Reynolds, G. (2011). *"**Principles of Information Systems**"* (2nd ed.). Australia: Cengage Learning.
doi:http://www.cengage.com/aushed/instructor.do?product_isbn=9780170188456

- Sleit, A., Hajaya, M., & Obisat, F. (2010). *"Video PowerSearcher: A Text-Based Indexing E-Learning System"*. ***ISWSA, International Conference on Intelligent Semantic Web - Services and Applications***.
doi: http://dl.acm.org/citation.cfm?id=1874613

- Thepade, S., & Tonge, A. (2014). *"An optimized key frame extraction for detection of near duplicates in content based video retrieval"*. ***IEEE, 2014 International Conference on Communications and Signal Processing (ICCSP)***, 1087 - 1091.
doi: 10.1109/ICCSP.2014.6950015

- Vrinda, & Shekhar, C. (2013). *"Speech Recognition System for English Language"*. ***IJARCCE, International Journal of Advanced Research in Computer and Communication Engineering***, 2(1), 919-922.
doi: http://www.ijarcce.com/volume-2-issue-1.html

- Welch, M., Cho, J., & Chang, W. (2010). *"Generating Advertising Keywords from Video Content"*. ***CIKM '10 Proceedings of the 19th ACM international conference on Information and knowledge management***, 1421-1424.
doi: 10.1145/1871437.1871637

- Xu, M., Xu, C., Duan, L., Jin, S., & Luo, S. (2008). *"Audio keywords generation for sports video analysis"*. ***ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)***, *4*(2).
doi: http://doi.acm.org/10.1145/1352012.1352015

- Zhang, Y., Xu, C., Rui, Y., Wand, J., & Lu, H. (2007). *"Semantic Event Extraction from Basketball Games using Multi-Modal Analysis"*. ***IEEE International Conference on Multimedia and Expo***, 2190-2193.
doi: 10.1109/ICME.2007.4285119
- Zheng, W., Li, J., Si, Z., Lin, F., & Zhang, B. (2006). *"Using high-level semantic features in video retrieval"*. ***CIVR'06 Proceedings of the 5th international conference on Image and Video Retrieval***, 370-379.
doi: 10.1007/11788034_38

Appendix

Speech Recognition Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OracleClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using KDXNet;
using System.Runtime.InteropServices;
using System.Speech.Recognition;
using System.Text.RegularExpressions;
using System.Linq;
using System.Web;

namespace KDXPlayerNet
{
    public partial class frmPlayer : Form
    {
        // Speech Recognition Define
        SpeechRecognitionEngine recognitionEngine;
        string[]  keyWords;
        Grammar g;
        public frmPlayer()
        {
            InitializeComponent();
            Initialize();
        }
        private void Initialize()
        {

            // prepare the grammer ..
            label2.Text = "sas.xls";
            string ConnectionString1 = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" + Application.StartupPath + "\\" + label2.Text + ";Extended
Properties=Excel 8.0";
            System.Data.OleDb.OleDbConnection ExcelConnection1 = new
System.Data.OleDb.OleDbConnection(ConnectionString1);

            string sqlQ = "Select text from [Sheet2$]";
             System.Data.OleDb.OleDbDataAdapter dt = new
System.Data.OleDb.OleDbDataAdapter(sqlQ, ExcelConnection1);
            // MessageBox.Show("" + dt.ToString() );
                DataSet ds = new DataSet();
                try
                {
                    ExcelConnection1.Open();
                    dt.Fill(ds);

                    keyWords = new string [ds.Tables[0].Rows.Count-1];
                    for (int i = 1; i < ds.Tables[0].Rows.Count; i++)
                    {
                        // To Avoid duplicate results ...
                         keyWords[i-1] = ds.Tables[0].Rows[i][0].ToString();
```

```
                }

                Choices chs = new Choices(keyWords);
                GrammarBuilder gb = new GrammarBuilder();
                gb.Append(chs);
                g = new Grammar(gb);
                g.Name = "SynoGrammer";

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                g = null ;
            }

        recognitionEngine = new SpeechRecognitionEngine();
        if(g != null)
          recognitionEngine.LoadGrammar(g);
       // else
        Grammar g2 = new DictationGrammar();
        g2.Name = "Dictation Grammer";
        recognitionEngine.LoadGrammar(g2);
        recognitionEngine.SetInputToDefaultAudioDevice();
        int c = 0;
        recognitionEngine.SpeechRecognized += (s, args) =>
        {
            foreach (RecognizedWordUnit word in args.Result.Words)
            {
                // You can change the minimun confidence level here
                c++;
                if (word.Confidence >= 0.75f)
                label2.Hide();
            }
        };
    }

private void btnTitleSearch_Click(object sender, EventArgs e)
    {
        txtSearch.Clear();
        try
        {
            lblState.Text = " Recognizing ... ";
            RecognitionResult result = recognitionEngine.Recognize(new
TimeSpan(0, 0, 2));
            if (result != null)
            {
                foreach (RecognizedWordUnit word in result.Words)
                {

                    txtSearch.Text += word.Text + " ";
                }

                txtSearch.Text = txtSearch.Text.Substring(0,
txtSearch.TextLength - 1);
                lblState.Text = " Done!  " + result.Grammar.Name;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
```

```
        }

        private void btnSynoSearch_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            try
            {
                lblState.Text = " Recognizing ... ";
                RecognitionResult result = recognitionEngine.Recognize(new
TimeSpan(0, 0, 2));
                if (result != null)
                {
                    foreach (RecognizedWordUnit word in result.Words)
                    {

                        textBox1.Text += word.Text + " ";
                    }

                    textBox1.Text = textBox1.Text.Substring(0,
textBox1.TextLength - 1);
                    lblState.Text = " Done! " + result.Grammar.Name;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```

Main Program Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OracleClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using KDXNet;
using System.Runtime.InteropServices;
using System.Speech.Recognition;
using System.Text.RegularExpressions;
using System.Linq;
using System.Web;

namespace KDXPlayerNet
{
    public partial class frmPlayer : Form
    {
        // Speech Recognition Define
        SpeechRecognitionEngine recognitionEngine;
        string[]  keyWords;
        Grammar g;
        public frmPlayer()
        {
            InitializeComponent();
```

```
            Initialize();
        }
        private void Initialize()
        {

            // prepare the grammer ..
            label2.Text = "sas.xls";
            string ConnectionString1 = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" + Application.StartupPath + "\\" + label2.Text + ";Extended
Properties=Excel 8.0";
            System.Data.OleDb.OleDbConnection ExcelConnection1 = new
System.Data.OleDb.OleDbConnection(ConnectionString1);

            string sqlQ = "Select text from [Sheet2$]";
             System.Data.OleDb.OleDbDataAdapter dt = new
System.Data.OleDb.OleDbDataAdapter(sqlQ, ExcelConnection1);
            // MessageBox.Show("" + dt.ToString() );
                DataSet ds = new DataSet();
                try
                {
                    ExcelConnection1.Open();
                    dt.Fill(ds);

                    keyWords = new string [ds.Tables[0].Rows.Count-1];
                    for (int i = 1; i < ds.Tables[0].Rows.Count; i++)
                    {
                        // To Avoid duplicate results ...
                         keyWords[i-1] = ds.Tables[0].Rows[i][0].ToString();

                    }

                    Choices chs = new Choices(keyWords);

                    GrammarBuilder gb = new GrammarBuilder();
                    gb.Append(chs);
                    g = new Grammar(gb);
                    g.Name = "SynoGrammer";

                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                    g = null ;
                }

            recognitionEngine = new SpeechRecognitionEngine();
            if(g != null)
              recognitionEngine.LoadGrammar(g);
           // else
            Grammar g2 = new DictationGrammar();
            g2.Name = "Dictation Grammer";
            recognitionEngine.LoadGrammar(g2);
            recognitionEngine.SetInputToDefaultAudioDevice();
            int c = 0;
            recognitionEngine.SpeechRecognized += (s, args) =>
            {
                foreach (RecognizedWordUnit word in args.Result.Words)
                {
                    // You can change the minimun confidence level here
                    c++;
                    if (word.Confidence >= 0.75f)
```

```
                label2.Hide();
            }
        };
    }

    // End of Speech Recognition
    private const int WM_COPYDATA = 0x4A;
    [StructLayout(LayoutKind.Sequential)]
    public struct COPYDATASTRUCT
    {
        public IntPtr dwData;
        public int cbData;
        public IntPtr lpData;
    }
    KDXNetMedia myMedia;
    int videoSize = 100;
    #region Constructor
    /// <summary>
    /// Constructor
    /// </summary>
    /// <param name="command"></param>
    ///

    public frmPlayer(string command)
    {
        InitializeComponent();
        Initialize();
        myMedia = new KDXNetMedia((Control)this.pictureBox1);
        myMedia.KDXMediaEvent += new KDXEventHandler(myMedia_KDXMediaEvent);
        myMedia.KDXTimerEvent += new
KDXTimerEventHandler(myMedia_KDXTimerEvent);
        myMedia.KDXSizeChangedEvent += new
KDXSizeChangeEventHandler(myMedia_KDXSizeChangedEvent);
        this.pictureBox1.ClientSizeChanged += new
EventHandler(pictureBox1_ClientSizeChanged);
        this.Slider.Scroll += new EventHandler(Slider_Scroll);
        this.AudioSlider.Scroll += new
ScrollEventHandler(AudioSlider_Scroll);
        this.mnuOpen.Click += new EventHandler(mnuOpen_Click);
        this.mnuExit.Click += new EventHandler(mnuExit_Click);
        this.mnuExit.ShortcutKeyDisplayString = "Ctrl + (X) [ESC]";
        this.mnuNormal.Click += new EventHandler(mnuNormal_Click);
        this.mnuNormal.ShortcutKeyDisplayString = "Ctrl + (N) [N]";
        this.mnuIncrease.Click += new EventHandler(mnuIncrease_Click);
        this.mnuIncrease.ShortcutKeys = (Keys)(Keys.Control | Keys.Add);
        this.mnuIncrease.ShortcutKeyDisplayString = "Ctrl + (+) [+]";
        this.mnuDecrease.ShortcutKeys = (Keys)(Keys.Control | Keys.Subtract);
        this.mnuDecrease.ShortcutKeyDisplayString = "Ctrl + (-) [-]";
        this.mnuDecrease.Click += new EventHandler(mnuDecrease_Click);
        this.mnuFullscreen.Click += new EventHandler(mnuFullscreen_Click);
        this.mnuFullscreen.ShortcutKeyDisplayString = "Ctrl + (F) [F]";
        this.DoubleBuffered = true;
        this.btnPlay.Click += new EventHandler(btnPlay_Click);
        this.btnPause.Click += new EventHandler(btnPause_Click);
        this.btnStop.Click += new EventHandler(btnStop_Click);
        this.mnuPlay.Click += new EventHandler(mnuPlay_Click);
        this.mnuStop.Click += new EventHandler(mnuStop_Click);
        this.mnuAudioIncrease.Click += new
EventHandler(mnuAudioIncrease_Click);
        this.mnuAudioDecrease.Click += new
EventHandler(mnuAudioDecrease_Click);
```

```
            this.mnuAudioSystem.Click += new EventHandler(mnuAudioSystem_Click);
            this.mnuAudioIncrease.ShortcutKeys = (Keys)(Keys.Alt | Keys.Add);
            this.mnuAudioIncrease.ShortcutKeyDisplayString = "Alt + (+) [Up]";
            this.mnuAudioDecrease.ShortcutKeys = (Keys)(Keys.Alt |
Keys.Subtract);
            this.mnuAudioDecrease.ShortcutKeyDisplayString = "Alt + (-) [Down]";
            this.mnuForward.Click += new EventHandler(mnuForward_Click);
            this.mnuForward.ShortcutKeyDisplayString = "Ctrl + (Right) [Right]";
            this.mnuFastForward.Click += new EventHandler(mnuFastForward_Click);
            this.mnuRewind.Click += new EventHandler(mnuRewind_Click);
            this.mnuRewind.ShortcutKeyDisplayString = "Ctrl + (LEFT) [LEFT]";
            this.mnuPlay.ShortcutKeyDisplayString = "Ctrl + (Space) [Space]";
            this.KeyDown += new KeyEventHandler(frmPlayer_KeyDown);
            btnPlay.Enabled = false;
            btnPause.Enabled = false;
            btnStop.Enabled = false;
            mnuSettings.Enabled = false;
            mnuAction.Enabled = false;
            if (command != "")
            {
                InitMedia(command);
            }
        }

        #endregion
        #region Local Events
        void frmPlayer_KeyDown(object sender, KeyEventArgs e)
        {
            if (myMedia.MediaPlayState != PlayState.Init)
            {
                switch (e.KeyCode.ToString())
                {
                    case "MediaPlayPause":
                        PlayPause();
                        break;
                    case "Space":
                        PlayPause();
                        break;
                    case "MediaStop":
                        Stop();
                        break;
                    case "MediaNextTrack":
                        mnuFastForward.PerformClick();
                        break;
                    case "MediaPreviousTrack":
                        mnuFastRewind.PerformClick();
                        break;
                    case "VolumeUp":
                        mnuAudioIncrease.PerformClick();
                        break;
                    case "VolumeDown":
                        mnuAudioDecrease.PerformClick();
                        break;
                    case "Escape":
                        this.Close();
                        break;
                    case "Left":
                        mnuRewind.PerformClick();
                        break;
                    case "Right":
                        mnuForward.PerformClick();
```

```
                    break;
                case "Up":
                    mnuAudioIncrease.PerformClick();
                    break;
                case "Down":
                    mnuAudioDecrease.PerformClick();
                    break;
                case "Add":
                    mnuIncrease.PerformClick();
                    break;
                case "Subtract":
                    mnuDecrease.PerformClick();
                    break;
                case "N":
                    mnuNormal.PerformClick();
                    break;
                case "F":
                    mnuFullscreen.PerformClick();
                    break;
                default:
                    break;
            }
        }
    }
    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        //Dont forget do Dispose the KDXNet object too free memory
        myMedia.Dispose();
        //Reclaim memory
        GC.Collect();
    }

    void mnuAssociate_Click(object sender, EventArgs e)
    {
        try
        {
            frmAssociate frm = new frmAssociate();
            frm.Show();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    void mnuFastRewind_Click(object sender, EventArgs e)
    {
        try
        {
            if (myMedia.MediaPlayState != PlayState.Init)
            {
                //Decrease the position with 60 seconds
                int x = Slider.Value - 60;
                //If that position is more than 0 seconds rewind
                if (x > 0)
                {
                    myMedia.SetCurrentPosition(x);
                }
            }
        }
        catch (Exception ex)
```

```
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuRewind_Click(object sender, EventArgs e)
        {
            try
            {

                if (myMedia.MediaPlayState != PlayState.Init)
                {
                    //Decrease the position with 10 seconds
                    int x = Slider.Value - 10;
                    //If that position is more than 0 seconds rewind
                    if (x > 0)
                    {
                        myMedia.SetCurrentPosition(x);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuFastForward_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.MediaPlayState != PlayState.Init)
                {
                    //Increase the position with 60 seconds
                    int x = Slider.Value + 60;
                    //If that position is not larger than the end
                    if (x < Slider.Maximum)
                    {
                        myMedia.SetCurrentPosition(x);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuForward_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.MediaPlayState != PlayState.Init)
                {
                    //Increase the position with 10 seconds
                    int x = Slider.Value + 10;
                    //If that position is not larger than the end
                    if (x < Slider.Maximum)
                    {
                        myMedia.SetCurrentPosition(x);
                    }
```

```
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuAudioSystem_Click(object sender, EventArgs e)
        {
            //Since I am way to lazy to program advanced audio setting
            //Open the systems Audiosettings :-)
            try
            {
                System.Diagnostics.Process proc = new
System.Diagnostics.Process();
                proc.EnableRaisingEvents = false;
                proc.StartInfo.FileName = "rundll32.exe";
                proc.StartInfo.Arguments = "shell32.dll Control_RunDLL
mmsys.cpl";
                proc.StartInfo.WindowStyle =
System.Diagnostics.ProcessWindowStyle.Normal;
                proc.Start();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuAudioDecrease_Click(object sender, EventArgs e)
        {
            try
            {
                int x = this.AudioSlider.Value;
                if (myMedia.MediaPlayState != PlayState.Init)
                {
                    // If current value is less or equal than 100% And more or
equal than 10%
                    //Decrease the volume with 10%
                    if (x <= 100 && x >= 10)
                    {
                        x -= 10;
                    }
                    this.AudioSlider.Value = x;
                    myMedia.SetVolume(this.AudioSlider.Value);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuAudioIncrease_Click(object sender, EventArgs e)
        {
            try
            {
                int x = this.AudioSlider.Value;
                if (myMedia.MediaPlayState != PlayState.Init)
                {
```

```
                    //If current volume is more than 90% set it to 100%
                    if (x > 90)
                    {
                        x = 100;
                    }
                    else
                    {
                        //Increase the volume in procent by 10
                        x += 10;
                    }
                    //Set it
                    this.AudioSlider.Value = x;
                    myMedia.SetVolume(this.AudioSlider.Value);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        void mnuStop_Click(object sender, EventArgs e)
        {
            try
            {
                Stop();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuPlay_Click(object sender, EventArgs e)
        {
            try
            {
                PlayPause();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuFullscreen_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.GetTypeOfMedia == ClipType.AudioVideo ||
myMedia.GetTypeOfMedia == ClipType.VideoOnly)
                {
                    myMedia.ToggleFullscreen();//Fullscreen on/off
                }
            }
            catch (Exception ex)
            {
```

```
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuDecrease_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.GetTypeOfMedia == ClipType.AudioVideo ||
myMedia.GetTypeOfMedia == ClipType.VideoOnly)
                {
                    videoSize -= 25;
                    myMedia.ResizeVideoWindow(videoSize);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuIncrease_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.GetTypeOfMedia == ClipType.AudioVideo ||
myMedia.GetTypeOfMedia == ClipType.VideoOnly)
                {
                    videoSize += 25;
                    myMedia.ResizeVideoWindow(videoSize);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuNormal_Click(object sender, EventArgs e)
        {
            try
            {
                if (myMedia.GetTypeOfMedia == ClipType.AudioVideo ||
myMedia.GetTypeOfMedia == ClipType.VideoOnly)
                {
                    videoSize = 100;
                    myMedia.ResizeVideoWindow(videoSize);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void mnuOpen_Click(object sender, EventArgs e)
        {
            try
            {
                OpenFileDialog openFileDialog = new OpenFileDialog();
                openFileDialog.Multiselect = false;
```

```
                openFileDialog.Filter = "Media
Files|*.mpg;*.avi;*.wma;*.mov;*.wav;*.mp2;*.mp3;*.divx;*.vob;*.dat|Video
Files|*.mpg;*.avi;*.wma;*.mov;*.mp2;*.divx;*.vob;*.dat|Audio
Files|*.wav;*.mp3|All Files|*.*";
                if (DialogResult.OK == openFileDialog.ShowDialog())
                {
                    //Initialize the media
                    InitMedia(openFileDialog.FileName);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        void btnStop_Click(object sender, EventArgs e)
        {
            try
            {
                Stop();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void btnPause_Click(object sender, EventArgs e)
        {
            try
            {
                PlayPause();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        void btnPlay_Click(object sender, EventArgs e)
        {
            try
            {
                PlayPause();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        void AudioSlider_Scroll(object sender, ScrollEventArgs e)
        {
            try
            {
                if (myMedia.MediaPlayState != PlayState.Init)
                {
                    if (e.NewValue > 90) { e.NewValue = 100; }
```

```
                if (e.NewValue < 10) { e.NewValue = 0; }
                myMedia.SetVolume(e.NewValue);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    void Slider_Scroll(object sender, EventArgs e)
    {
        try
        {
            if (myMedia.MediaPlayState != PlayState.Init)
            {
                TimeSpan ts = new TimeSpan(0, 0, Slider.Value);
                TimeSpan ts1 = new TimeSpan(0, 0, Slider.Maximum);
                this.lblTid.Text = ts.Hours.ToString("00") + ":" +
ts.Minutes.ToString("00") + ":" + ts.Seconds.ToString("00") + "/" +
ts1.Hours.ToString("00") + ":" + ts1.Minutes.ToString("00") + ":" +
ts1.Seconds.ToString("00");
                myMedia.SetCurrentPosition(Slider.Value);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    void pictureBox1_ClientSizeChanged(object sender, EventArgs e)
    {
        //Resize this, Accordingly to the videowindow, take into account
other controls, position it in the middle of screen
        try
        {
            if (this.WindowState == FormWindowState.Normal)
            {
                //this.ClientSize = pictureBox1.ClientSize;
                //this.Height = pictureBox1.ClientSize.Height +
this.panel1.ClientSize.Height + this.menuStrip1.ClientSize.Height + 25;
                //this.Top = (Screen.PrimaryScreen.WorkingArea.Height -
this.ClientSize.Height) / 2;
                //this.Left  = (Screen.PrimaryScreen.WorkingArea.Width  -
this.ClientSize.Width) / 2;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "KDXPlayerNet", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    #endregion
    #region KDXNet Events
    void myMedia_KDXTimerEvent(string currenttime, string totaltime, int
totalseconds)
    {
```

```
            this.lblTid.Text = currenttime + "/" + totaltime; Slider.Value = 
totalseconds;
        }
        void myMedia_KDXMediaEvent(PlayState playstate)
        {
            if (playstate == PlayState.Running)
            {
                //Set the slider to total no of seconds of the media
                Slider.Minimum = 0;
                Slider.Maximum = myMedia.MediaTimeTotalSeconds;
                //Show the name of the media
                this.Text = myMedia.MediaName;
            }
            else if (playstate == PlayState.Stopped)
            {
                mnuPlay.Text = "Play";
                btnPlay.Enabled = true;
                btnPause.Enabled = false;
            }
        }
        void myMedia_KDXSizeChangedEvent(Size Size)
        {
            this.pictureBox1.Size = Size;
        }
        #endregion

        #region Action
        void InitMedia(string mediafilepath)
        {
            try
            {
                if (File.Exists(mediafilepath))
                {
                    //Reclaim memory
                    GC.Collect();
                    //Initialize the media
                    myMedia.OpenVideoFile(mediafilepath);
                    //Find out what type of media it is, Enable/disable menues 
and buttons accordingly
                    if (myMedia.GetTypeOfMedia == ClipType.AudioOnly)
                    {
                        mnuSize.Enabled = false;
                        pictureBox1.ClientSize = new System.Drawing.Size(260, 0);
                    }
                    else
                    {
                        mnuSize.Enabled = true;
                    }
                    if (myMedia.GetTypeOfMedia == ClipType.VideoOnly)
                    {
                        mnuAudio.Enabled = false;
                        AudioSlider.Visible = false;
                    }
                    else
                    {
                        mnuAudio.Enabled = true;
                        AudioSlider.Visible = true;
                        myMedia.SetVolume(this.AudioSlider.Value);
                    }
                    mnuPlay.Text = "Play";
                    btnPlay.Enabled = true;
```

```
                btnPause.Enabled = false;
                btnStop.Enabled = true;
                mnuSettings.Enabled = true;
                mnuAction.Enabled = true;
                //Start playback
                PlayPause();
                this.Focus();
            }
        }
        catch (Exception ex)
        {
            //throw ex;
        }
    }
    void Stop()
    {
        try
        {

            myMedia.Stop();
            mnuPlay.Text = "Play";
            btnPlay.Enabled = true;
            btnPause.Enabled = false;
            Slider.Value = 0;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
    void PlayPause()
    {
        try
        {
            if (mnuPlay.Text == "Play")
            {
                myMedia.Play();
                mnuPlay.Text = "Pause";
                btnPlay.Enabled = false;
                btnPause.Enabled = true;
            }
            else
            {
                myMedia.Pause();
                mnuPlay.Text = "Play";
                btnPlay.Enabled = true;
                btnPause.Enabled = false;
            }
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
    #endregion

    #region Messagehandler
    /// <summary>
    /// Messagehandler, listen for windows messages
    /// </summary>
    /// <param name="m"></param>
```

```
        protected override void WndProc(ref Message m)
        {
            switch (m.Msg)
            {
                case WM_COPYDATA:
                    string strCmdLine = "";
                    COPYDATASTRUCT cds = new COPYDATASTRUCT();
                    cds = (COPYDATASTRUCT)Marshal.PtrToStructure(m.LParam,
typeof(COPYDATASTRUCT));
                    if (cds.cbData > 0)
                    {
                        strCmdLine =
Marshal.PtrToStringAnsi(cds.lpData).Substring(0, cds.cbData);
                        //if (strCmdLine.LastIndexOf(" ") > 0)
                        //{
                        //    strCmdLine =
strCmdLine.Substring(strCmdLine.LastIndexOf(" ") + 1).Trim();
                        //}
                        if (strCmdLine.Trim().Length > 0)
                        {
                            // Play the media passed as commandline arg to second
instance
                            this.InitMedia(strCmdLine.Replace("\"",
"").Replace(Application.ExecutablePath, "").Trim());
                        }
                        // Bring to front
                        this.Activate();
                    }
                    break;
            }
            base.WndProc(ref m);
        }
        #endregion

        private void frmPlayer_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'myDbDataSet.Videos'
table. You can move, or remove it, as needed.
            // this.videosTableAdapter.Fill(this.myDbDataSet.Videos);
            dataGridView3.Visible = false;
        }

        private void dataGridView2_Click(object sender, EventArgs e)
        {
            try
            {
                GC.Collect();
                myMedia.OpenVideoFile(Application.StartupPath + "\\" +
myDbDataSet.Videos.Rows[dataGridView2.CurrentCell.RowIndex]["VideoPath"].ToString
());
                myMedia.Play();
                label1.Text =
myDbDataSet.Videos.Rows[dataGridView2.CurrentCell.RowIndex]["ExcelPath"].ToString
();
                label2.Text =
myDbDataSet.Videos.Rows[dataGridView2.CurrentCell.RowIndex]["SynoPath"].ToString(
);
                myMedia.Pause();
                btnPlay.Enabled = true;
                btnPause.Enabled = false;
                btnStop.Enabled = false;
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            dataGridView1.Visible = true;
            dataGridView3.Visible = false;
            if (textBox1.TextLength == 0)
            {
                MessageBox.Show("Please type a search keyword in the box");
            }
            else
            {
                string ConnectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" + Application.StartupPath + "\\" + label1.Text + ";Extended
Properties=Excel 8.0";
                System.Data.OleDb.OleDbConnection ExcelConnection = new
System.Data.OleDb.OleDbConnection(ConnectionString);

                string ConnectionString1 =
@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application.StartupPath + "\\"
+ label2.Text + ";Extended Properties=Excel 8.0";
                System.Data.OleDb.OleDbConnection ExcelConnection1 = new
System.Data.OleDb.OleDbConnection(ConnectionString1);

                string SynQuery;
                string ExcelQuery;


                ExcelQuery = "SELECT * FROM [Sheet1$] where Text like '%" +
textBox1.Text + "%'";

                System.Data.OleDb.OleDbDataAdapter dt = new
System.Data.OleDb.OleDbDataAdapter(ExcelQuery, ExcelConnection);

                DataSet ds = new DataSet();
                try
                {
                    ExcelConnection.Open();
                    ExcelConnection1.Open();
                    dt.Fill(ds);



                    dsVideo.Results.Clear();
                    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                    {
                        // To hide duplicate data ...
                        dsVideo.Results.AddResultsRow(textBox1.Text + "   " + (i +
1).ToString(), ds.Tables[0].Rows[i][3].ToString());
                    }

                        // Add extra Synonum ...
                    SynQuery = "SELECT * FROM [Sheet2$] WHERE id in ( select id
from [Sheet2$]  where text = '" + textBox1.Text + "')";
                    System.Data.OleDb.OleDbDataAdapter dtSyn = new
System.Data.OleDb.OleDbDataAdapter(SynQuery, ExcelConnection1);
```

```
                    DataSet dsSyn = new DataSet();
                    dtSyn.Fill(dsSyn);
                    for (int j = 0; j < dsSyn.Tables[0].Rows.Count; j++)
                    {
                        ExcelQuery = "SELECT * FROM [Sheet1$] where Text like '%"
+ dsSyn.Tables[0].Rows[j][1].ToString() + "%'";
                        System.Data.OleDb.OleDbDataAdapter dt2 = new
System.Data.OleDb.OleDbDataAdapter(ExcelQuery, ExcelConnection);

                        DataSet ds2 = new DataSet();

                        //ExcelConnection.Open();
                        dt2.Fill(ds2);

                        //dsVideo.Results.Clear();
                        for (int k = 0; k < ds2.Tables[0].Rows.Count; k++)
                        {
                            if (dsSyn.Tables[0].Rows[j][1].ToString() !=
textBox1.Text)
                            {

dsVideo.Results.AddResultsRow(dsSyn.Tables[0].Rows[j][1].ToString() + "  " + (k +
1).ToString(), ds2.Tables[0].Rows[k][3].ToString());
                            }
                        }
                    }
                }
                catch (Exception ex)
                {
                    ExcelConnection.Close();
                    MessageBox.Show(ex.Message);
                }
            }
        }

        private void btnSearchForVideos_Click(object sender, EventArgs e)
        {
            if (txtSearch.TextLength == 0)
            {
                MessageBox.Show("Please type a search keyword in the box");
            }
            else
            {
                DataAccess cls = new DataAccess();
                string sql = "select * from Videos where Keyword like '%" +
txtSearch.Text + "%' or VideoName like '%" + txtSearch.Text + "%'";
                myDbDataSet.Videos.Clear();
                string m = cls.getDataTable(sql, ref myDbDataSet, "Videos");
            }
        }

        private void dataGridView1_Click(object sender, EventArgs e)
        {
            try
            {
                Slider.Value =
Convert.ToInt32(decimal.Round(Convert.ToDecimal(dsVideo.Results.Rows[dataGridView
1.CurrentCell.RowIndex]["StartTime"]), 0));
                Slider_Scroll(sender, e);
                btnPause.Enabled = true;
                btnPlay.Enabled = false;
```

```
                btnStop.Enabled = true;
                myMedia.Play();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

      //Start
        private void btnstop1_Click(object sender, EventArgs e)
        {
            // Stop ( Button_Click )
            // recognitionEngine.RecognizeAsyncStop();
            // startButton.Enabled = true;
            // btnstop1.Enabled = false;
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

            string ConnectionString1 = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" + Application.StartupPath + "\\" + label1.Text + ";Extended
Properties=Excel 8.0";
            System.Data.OleDb.OleDbConnection ExcelConnection1 = new
System.Data.OleDb.OleDbConnection(ConnectionString1);

            string SynQuery;

            SynQuery = "SELECT Text FROM [Sheet1$] where lcase(Text) like ('%" +
textBox1.Text + "%')";
            System.Data.OleDb.OleDbDataAdapter dt = new
System.Data.OleDb.OleDbDataAdapter(SynQuery, ExcelConnection1);

            DataSet ds = new DataSet();
            try
            {
                ExcelConnection1.Open();
                dt.Fill(ds);

                dsVideo.Results.Clear();
                for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                {
                    var fixedInput =
Regex.Replace(ds.Tables[0].Rows[i][0].ToString(), "[^a-zA-Z0-9% ._]",
string.Empty);
                    // Remove all special characters:
                    // This regex doesn't support apostrophe so the extension
method is better
                    var splitted = fixedInput.Split(' ');
                    // Split Array:
                    string[] stringArray = splitted;
                    if (textBox1.Text.Contains(" "))
                    {
                        dsVideo.Results.AddResultsRow(ds.Tables[0].Rows[i][0] +
"", ds.Tables[0].Rows[i][0].ToString());
                    }
                    else
                    {
                        var index1 = Array.FindIndex(stringArray, row =>
row.Contains(textBox1.Text));
```

```
                        // Find the index of searched text:
                        if ((splitted.GetValue(index1) + "") != null)
                        {
                            // To hide duplicate data ...

dsVideo.Results.AddResultsRow(splitted.GetValue(index1) + "",
ds.Tables[0].Rows[i][0].ToString());
                        }
                        else
                        {
                            return;
                        }
                        //>>>
                        try
                        {
                            if (dataGridView3.Rows.Count > 1)
                            {

                                for (int currentRow = 0; currentRow <
dataGridView3.Rows.Count; currentRow++)
                                {
                                    var rowToCompare =
dataGridView3.Rows[currentRow]; // Get row to compare against other rows

                                    // Iterate through all rows
                                    //
                                    foreach (DataGridViewRow row in
dataGridView3.Rows)
                                    {
                                        if (rowToCompare.Equals(row))
                                        {
                                            continue;
                                        }
                                        // If row is the same row being compared,
skip.

                                        bool duplicateRow = true;

                                        // Compare the value of all cells
                                        //

                                        if (rowToCompare.Cells[0].Value != null
&& rowToCompare.Cells[0].Value.ToString() != row.Cells[0].Value.ToString())
                                        {
                                            duplicateRow = false;

                                        }


                                        if (duplicateRow)
                                        {

dataGridView3.Rows.RemoveAt(row.Index);
                                            break;
                                        }
                                    }
                                }
                            }
                        }
                        catch
                        {
```

```
                    }
                    //>>>
                }
            }

         }

    //Stop
            catch (Exception ex)
            {
           //     ExcelConnection1.Close();
             //   MessageBox.Show(ex.Message);
            }
            dataGridView3.Visible = true;
            dataGridView1.Visible = false;
        }

        private void dataGridView3_CellContentClick_1(object sender,
DataGridViewCellEventArgs e)
        {
            textBox1.Text =
Convert.ToString(dsVideo.Results.Rows[dataGridView3.CurrentCell.RowIndex][0]);
            dataGridView3.Visible = false;
            dataGridView1.Visible = true;
            dsVideo.Results.Clear();

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            label3.Left -= 1; if (label3.Left == (0 - label3.Width)) {
label3.Left = this.Width; }
        }

        private void btnstop1_Click_1(object sender, EventArgs e)
        {
            recognitionEngine.RecognizeAsyncStop();
        }

        private void btnTitleSearch_Click(object sender, EventArgs e)
        {
            txtSearch.Clear();
            try
            {
                lblState.Text = " Recognizing ... ";
                RecognitionResult result = recognitionEngine.Recognize(new
TimeSpan(0, 0, 2));
                if (result != null)
                {
                    foreach (RecognizedWordUnit word in result.Words)
                    {

                        txtSearch.Text += word.Text + " ";
                    }

                    txtSearch.Text = txtSearch.Text.Substring(0,
txtSearch.TextLength - 1);
                    lblState.Text = " Done! " + result.Grammar.Name;
                }
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnSynoSearch_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            try
            {
                lblState.Text = " Recognizing ... ";
                RecognitionResult result = recognitionEngine.Recognize(new
TimeSpan(0, 0, 2));
                if (result != null)
                {
                    foreach (RecognizedWordUnit word in result.Words)
                    {

                        textBox1.Text += word.Text + " ";
                    }

                    textBox1.Text = textBox1.Text.Substring(0,
textBox1.TextLength - 1);
                    lblState.Text = " Done! " + result.Grammar.Name;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void IDTDB_Click_1(object sender, EventArgs e)
        {
            //replace "MyNewForm" with the class name of the form
            Form form1 = new Form1();
            form1.Show();
        }

        private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void AM_Click(object sender, EventArgs e)
        {
            //replace "MyNewForm" with the class name of the form
            Form form2 = new AboutBox1();
            form2.Show();
        }
        }
}
```

Program.cs Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using System.Diagnostics;
using System.Runtime.InteropServices;
using System.Speech.Recognition;

namespace KDXPlayerNet
{
    static class Program
    {
        private const int WM_COPYDATA = 0x4A;
        [StructLayout(LayoutKind.Sequential)]
        public struct COPYDATASTRUCT
        {
            public IntPtr dwData;
            public int cbData;
            public IntPtr lpData;
        }
        [DllImport("user32.dll")]
        public static extern long SendMessage(
            IntPtr hWnd,
            uint Msg,
            uint wParam,
            ref COPYDATASTRUCT lParam
            );
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            bool blnstart = true;
            string command =
System.Environment.CommandLine.Replace("\"","").Replace(Application.ExecutablePat
h,"").Trim();
          //  MessageBox.Show("Command = " + command);
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Process p = Process.GetCurrentProcess();
            Process[] processes = Process.GetProcessesByName(p.ProcessName);
            if (processes.Length != 1)
            {
                foreach (Process proc in processes)
                {
                    if (proc.MainWindowHandle  != p.MainWindowHandle)
                    {
                        SendMessageToRunning(proc.MainWindowHandle);
                        blnstart = false;
                        break;
                    }
                }
            }
            if (blnstart) { Application.Run(new frmPlayer(command)); }
        }
        private static void SendMessageToRunning(IntPtr hwnd)
        {
            IntPtr ptr = Marshal.StringToHGlobalAnsi(Environment.CommandLine);
```

```
            COPYDATASTRUCT cds = new COPYDATASTRUCT();
            cds.dwData = IntPtr.Zero;
            cds.cbData = Environment.CommandLine.Length;
            cds.lpData = ptr;

            long result = SendMessage(hwnd, WM_COPYDATA, 0, ref cds);

            // Required to free unmanaged memory otherwise a memory leak occurs
            Marshal.FreeHGlobal(ptr);
        }
    }
}
```