Adaptive e-Learning AI-Powered Chatbot based on Multimedia Indexing

Salma El Janati[1], Abdelilah Maach[2], Driss El Ghanami[3]
LRIE Laboratory, Mohammadia School of Engineers (EMI)
Mohammed V University, Rabat
Morocco

Abstract—**With the rapid evolution of e-learning technology, the multiple sources of information become more and more accessible. However, the availability of a wide range of e-learning offers makes it difficult for learners to find the right content for their training needs. In this context, our paper aims to design an e-learning AI-powered Chatbot allowing interaction with learners and suggesting the e-learning content adapted to their needs. In order to achieve these objectives, we first analysed the e-learning multimedia content to extract the maximum amount of information. Then, using Natural Language Processing (NLP) techniques, we introduced a new approach to extract keywords. After that, we suggest a new approach for multimedia indexing based on extracted keywords. Finally, the Chatbot architecture is realized based on the multimedia indexing and deployed on online messaging platforms. The suggested approach aims to have an efficient way to represent the multimedia content based on keywords. We compare our approach with approaches in literature and we deduce that the use of keywords on our approach result on a better representation and reduce time to construct multimedia indexing. The core of our Chatbot is based on this indexed multimedia content which enables it to look for the information quickly. Then our designed Chatbot reduce response time and meet the learner's need.**

Keywords—e-Learning; Chatbot; Speech-To-Text; NLP; Keywords Extraction; Text Clustering; Multimedia Indexing

I. Introduction

With the rapid evolution of the IT field and the continuous updating of available tools. Learners must undergo continuous trainings in order to improve their skills and ensure technological monitoring [1]. Indeed, training has been strongly affected by the digital transformation. e-Learning (online training) is a perfect example with the digitization of learning. This distance learning technique eliminates the physical presence of a trainer [2]. It has many advantages and is become a part of the learner journey. However, the quantity of multimedia contents offered for the learner increases exponentially, which makes the autonomous learning a complicated task because it is necessary to choose the content adapted to their context to ensure an effective learning [3].

In this context, our study aims to create an indexed database of e-learning multimedia content in order to set up a Chatbot system who offer the appropriate content to each learner according to his or her e-learning needs. The advancements in artificial intelligence and NLP, allowing bots to converse more and more, like real people who share e-learning presentation contents. Predominant Chatbots do not depend exclusively on content, and will frequently appear valuable cards, pictures, joins, and shapes, giving an app-like encounter.

One of the most difficult areas of research is the development of effective Chatbots that emulate human dialogue. It is a difficult task that involves problems related to the NLP (Natural Language Processing) research field [4]. Thanks to NLP techniques and algorithms, it possible to understand the learners' requests based on what the learner is writing. Usually, this task is the core of the Chatbot but there are some limitations as it is not possible to map all learner requests, and current Chatbots do not show remarkable performance due to the unpredictability of the thinking of the learner during a conversation [5]. One of the most important task in setting up Chatbot is the design of the conversational flow. In fact, we suggest a new approach for a successful conversation based on keywords extractions, which it is important to handle with all learners requests and provide the adequate content.

This paper is structured as follows: the second section is a related work. In the third section, we will present the suggested approach. Then, the fourth part will be devoted to our result and our simulation. Finally, we will conclude with a recommendation approach to improve the suggested approach.

II. Related Work

The use of the advanced technology in data science enables to improve the quality of e-learning content. In this paper, we suggest a framework that uses Natural Language Processing (NLP) and Keywords Extraction as a chatbot engine for e-learning. Thus, in this section we will review the literature related to chatbot and Automatic Keywords Extraction (AKE).

A. Chatbot

Chatbots are a virtual assistant capable of chatting with users and responding to their requests. They are increasingly using speech synthesis techniques to produce their messages as the user types their interventions into a text field on the web page.

The design of a Chatbot to meet the needs of users was always a concern in the field of information retrieval [4]. Depending on how Chatbots are programmed, we can divide them into two large groups: those that are programmed according to predefined commands (rule-based Chatbot) [5] and those based on artificial intelligence (AI) [6].

AI Chatbots using machine learning are designed to understand the context and intent of a question before formulating an answer. There are two types of AI Chatbots: Generative Chatbots [7] and Information Retrieval Chatbots [4]. Fig. 1 presents Chabot's types.

The generative Chatbot is built in order to be able to act to any context or interlocutor and also in nonprogrammed situations. Such a conversational agent relies on the new artificial intelligence techniques such as deep learning and neural network to generate his responses word by word [8]. Thus, these bots can construct answers to users questions themselves. The problem with this approach is that it is too general and Chatbots struggle to have a coherent conversation or even to produce syntactically valid sentences with current models.

The Information retrieval Chatbots are the chatbot adapted to a given context, which builds its responses using a set of sentences that have been given to it in advance. These chatbots are based on retrieving information from the user's question and seek the most suitable answer using NLP [9]. This type of Chatbots is best suited for closed domain systems. This approach guarantees grammatically correct answers and simplifies the learning task for the algorithm, because it allows constructing the model on a training data set smaller than the big amount of data required in the case of a generative Chatbot [4].

Thus, in our case study we chose to design a Chatbot based on retrieval information since it allows to exploit the results obtained from videos indexing and also have some control over the responses generated by the Chatbot.

Our contribution consists of using a keyword extraction technique adapted to the multimedia e-learning content we have. Thus, the Chatbot will be based on the keywords instead of all the text in order to find the adequate answer to the learner's need in a fast and efficient way. The next section will be dedicated to keyword extraction techniques found in literature.

B. Automatic Keywords Extraction (AKE)

Keyword extraction involves identifying the words and phrases representing the main subjects of a document. High quality keywords can make it easier to understand, organize and access the content of the document. AKE from a document has been used in many applications, such as information retrieval [10], text synthesis [11], text categorization [12], and opinions mining [13].

Most existing keyword extraction algorithms address this problem in three steps (Fig. 2): First, the candidate keywords (i.e. words and phrases that can be used as keywords) are selected in the content of the document. Second, candidates are either classified using a candidate weighting function (unsupervised approaches) or classified into two classes (keywords / no keywords) using a set of extracted characteristics (supervised approaches). Third, the most weighted first N candidates with the highest confidence scores are selected as keywords [14].

In this section, we provide an overview of keyword extraction methods. Our goal is not to detail the functioning of these methods, but rather to have an overview of their basic principle and their classification in order to identify the suitable methods for our case study.

1) Statistic approach: Statistical Approaches is considered one of the simplest techniques used to identify keywords within a text. These approaches do not require training data in order to extract the most important keywords in a text. it seeks to define what a keyword, based on certain statistical features and study their relation with the notion of importance of a candidate term. the more the term candidate is considered important in analysed document, and the more it will be relevant as a keyword.

TF-IDF [15] and Likey [16] are two methods, which compare the behavior of a candidate term in the analysed document with its behavior in a collection of documents (reference corpus). The objective is to find candidate terms whose behavior in the document varies positively compared to their overall behavior in the collection. In both methods, this is expressed by the fact that a term is important in the document analysed if it is largely present, when it is not in the rest of the collection.

Yake! Approach [17] focuses on statistical features that do not require external dictionaries and look on characteristics, which can be calculated using only current document. These approaches are based on characteristics such as the position of the first occurrence of a candidate, the word frequency, the case, and the frequency with which a word appears in different sentences.

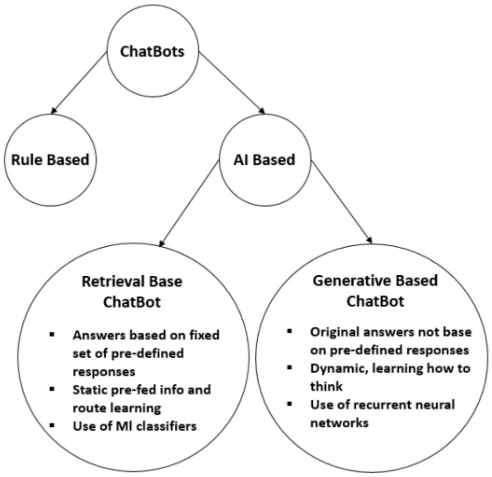

ChatBots
Rule Based
AI Based
Retrieval Base ChatBot
Answers based on fixed set of pre-defined responses
Static pre-fed info and route learning
Use of Ml classifiers
Generative Based ChatBot
Original answers not base on pre-defined responses
Dynamic, learning how to think
Use of recurrent neural networks

Fig. 1. Chatbots Types.

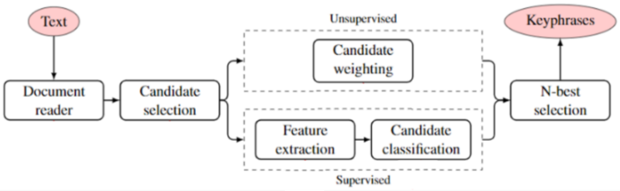


Fig. 2. General approach for Automatic Keywords Extraction.

2) Graph based approach: Graph-based approaches consist of representing the content of a document as a graph. The methodology applied comes from PageRank [18], an algorithm for ranking web pages (graph nodes) based on the recommendation links that exist between them (graph edges).

TextRank [19] and SingleRank [20] are the two basic adaptations of PageRank for AKE. In these, web pages are replaced by text units whose granularity is the word and an edge is created between two nodes if the words they represent co-occur in a given word window. When running the algorithm, a score is associated with each candidate keyword, which represents the importance of the node in the graph.

To improve TextRank / SingleRank, Liu et al. suggest a method, which aims to increase the coverage of all the key terms extracted in the analysed document (TopicalPageRank) [21]. To do this, they try to refine the importance of words in the document by taking into account their rank in each topic. The rank of a word for a topic is obtained by integrating into its PageRank score the probability that it belongs to the topic. The overall rank of a candidate term is then obtained by merging its ranks for each topic.

3) Word embedding approach: Word embedding [22] is a new way of representing words as vectors, typically in a space of a few hundred dimensions. A word is transformed into numbers. The word representation vector is learned based on an iterative algorithm from a large amount of text. The algorithm tries to put the vectors in space in order to bring together the semantically close words, and to move away the semantically distant words. By finding the closest words in the embedding space of a given word as input, the model identifies synonyms or intruders in a list of words. Once a model is obtained, several standard tasks become possible.

Different methods based on the word embedding are suggested for representing entire documents or sentences [23]. Skip-thought [24] provides sentence embedding trained to predict neighbouring sentences. Sent2Vec [25] generate sentence embedding using word n-gram representation.

With words embedding the keyword can be extracted using the cosine similarity in the word space representation. EmbedRank [26] is a word embedding based approach to automatically extract key phrases. In this method, documents or word sequences of arbitrary length are embedded into the same feature space. This enables computing semantic relatedness between document and candidate keyword by using the cosines similarity measures.

4) Supervised approach: Supervised approaches are methods who able to learn to perform a particular task, in this case the extraction of keywords. Learning is done through a corpus whose documents are annotated in keywords. The annotation allows to extract the examples and counter-examples whose statistical and / or linguistic features are used to teach a binary classification [27]. These classifications consist of indicating if a candidate term is a keyword or not. Many supervised algorithms are used in various fields. They can adapt to any task, including AKE task. Algorithms used for this construct probabilistic models, decision trees, Support Vector Machine (SVM) or even neural networks.

KEA is a method which uses a naive Bayesian classification to assign a likelihood score to each candidate term, the aim being to indicate whether they are keywords or not [28]. These approaches use three conditional distributions learned from the learning corpus. The first is the probability that each candidate term is labelled yes (keyword) or no (no keyword). The other two stand for two different statistic features which are the TF-IDF weight of the term candidate and its first position in the document.

Nguyen et al. propose an improvement (WINGNUS) [29] by adding a set of features such as: First and last occurrences (word off-set), Length of phrases in words, whether a phrase is also part of the document title, number of times a phrase appears in the title of other document. Adding these features improves the performance of the original version of KEA, but only when the amount of data is large enough. Suggested Architecture.

III. Suggested Approach

In this section, we suggest an approach of an adaptive Chatbot based on Retrieval information. This approach consists of indexing e-learning multimedia using keyword extraction. These indexed contents will be integrated in the Chatbot engine in order to offer the adequate content to the learner's needs.

The suggested approach is based on four steps, which will be detailed:

1) Extracting metadata from the e-learning database.
2) Speech to text Processing.
3) Automatic Keywords Extraction.
4) Suggested Chatbot design.

The first step of our approach consists of analyzing e-learning multimedia content and extracting as much information as possible from e-learning content in order to build our database. The second step will be devoted to standardize the e-learning content by transforming all content to text, so we use the Speech To Text techniques in order to extract text from all multimedia content. In the third step, we will suggest methods for extracting keywords from the extracted text. We will test several approaches and suggest a new approach adapted to our problematic.The last step describe chatbot design based on keywords resulting from the previous steps. Fig. 3 summarizes the methodology of our approach.

A. Extracting Metadata from e-Learning Database

In our study, we use a various source of e-learning content in order to construct database, which will be the base of our Chatbot recommendation. e-Learning sources provide content in different types of information like video, speech and, text. The first step of our approach consists of extracting as much information as possible from e-learning content in order to build our database.

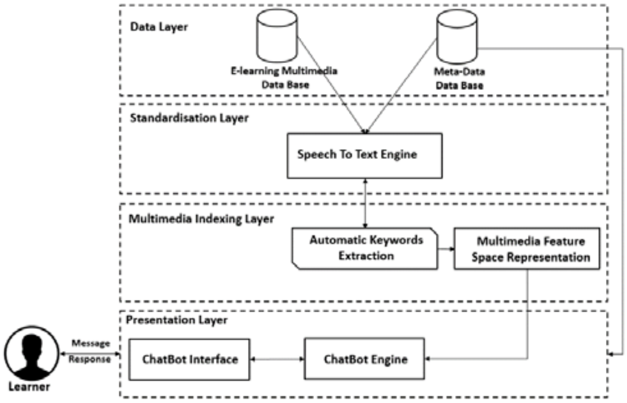


Fig. 3. Methodology Flow Chart.

In order to extract metadata from e-learning multimedia, we design a Python script that automatically retrieves metadata. This data frame is rich in information, it mainly contains: ID, the title, the category, the description, the subtitle if it is +available, the date of the online publication, and the author. In addition, the metadata contains the audio of multimedia if it is a video content. This metadata information is used in the Chabot engine in order to organize multimedia e-learning content and to give an easy content access. Fig. 4 shows the metadata extraction script.

The second step of our approach is to standardize the e-learning content by transforming all types into text (video to text and speech to text). The next sub-section is dedicated to describe the speech to text approach used in our case study.

B. Speech to Text Processing

Automatic Speech Recognition (ASR) is the process by which speech is transcribed into text. This technology has many useful applications ranging from hands-free car interfaces to home automation. Although speech recognition is an easy task for humans, it has always been difficult for machines. Since 2012, the use of Deep Neural Networks (DNN) has considerably improved the accuracy of speech recognition [30].

Deep Learning-based ASR systems today have achieved even better results than humans in languages like English [Baidu's Deep speech 2] [31]. These advances have been propelled by the use of large amounts of data (up to tens of thousands of hours of transcribed speech) and by an enormous parallel computing power controlled by GPUs.

The general approach of ASR systems based on Deep Learning and which is currently used by almost all systems offered by large groups such as google, IBM ... This approach follows the architecture presented in Fig. 5.

This approach generally starts from converting audio into a feature matrix to feed it into the neural network. This is done by creating spectrograms from audio waveform. The spectrogram input can be considered as a vector at each timestamp. A 1D convolutional layer extricates highlights out of each of these vectors to provide a grouping of highlight vectors for the LSTM layer to handle. The (Bi) LSTM output layer is passed to a Fully Connected layer for each time step which gives a probability distribution of the character at that time step using softmax activation.

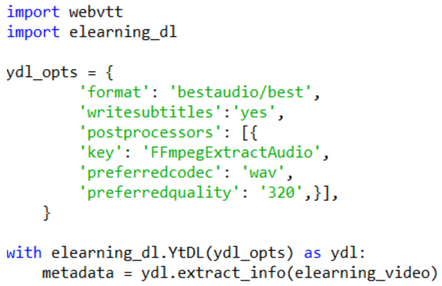
The problem of automatically recognizing speech with the assistance of a computer is a difficult task, due to the complexity of the human dialect. To solve this problem, several challenges must be addressed: Microphone poor quality, background noise, speaker variability and so on. All these possibilities must be included in the training step for Deep Network to process them. Thus, to create a voice-recognition system that achieves the performance of Siri, Google Now or Alexa, it is mandatory to have a large amount of data for the training step.

Given the difficulty of acquiring a massive database to train our own ASR system, we chose in our study to evaluate the APIs that are available on the market. Thus, we develop a Python Script that implement different Speech Recognition API's using 'Speech Recognition' library.

In order to compare different methods proposed by API's for automatic transcription, we need to evaluate the performance of each model. The key metric for transcription is accuracy. How closely the words within the created transcript match the talked words within the unique sound.

To calculate the accuracy of the automatic transcription, we will use the metric Word Error Rate (WER). The WER is a very simple and widely use measure for transcription accuracy. It is a number, calculated as the number of words needed to be inserted or changed or deleted to convert the transcript hypothesis into the reference transcript, divided by the number of words in the reference transcript. (It's the Levenshtein distance for words, measuring the minimum number of single words edits to correct the transcription.) A perfect match has a WER of zero; larger values indicate lower accuracy and thus more editing.

$$Word\ Error\ Rate = \frac{Insertions + Deletions + Substitutions}{Number\ of\ Words\ in\ Reference\ Transcript}$$



```
import webvtt
import elearning_dl

ydl_opts = {
        'format': 'bestaudio/best',
        'writesubtitles':'yes',
        'postprocessors': [{
        'key': 'FFmpegExtractAudio',
        'preferredcodec': 'wav',
        'preferredquality': '320',}],
    }

with elearning_dl.YtDL(ydl_opts) as ydl:
    metadata = ydl.extract_info(elearning_video)
```

Fig. 4. Metadata Extraction Script.

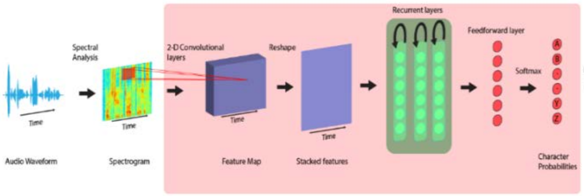


Fig. 5. Global Architecture of ASRs based on Deep Learning.

Fig. 6 and Table I show WER evaluation obtained from the APIs transcription compared to ground truth.

The results obtained show that the transcription of YouTube is the most efficient, followed by the transcription of Houndify. This is because the YouTube transcription is generally based on the publisher's recommendation, but this transcription is not always available. Houndify API specializes in detecting lyrics in music and that can explain the good performance of this API, since our e-learning data source mainly contains a speech with a music background. We can also notice that IBM and Google services have a similar performance since they are based on similar resources for learning phase. Finally, the Wit transcription is the least efficient and this is generally due to resources for the learning phase which are quite limited.

Based on results obtained is this step, we decided to make the approach shown on Fig. 7, in order to normalize multimedia e-learning types and extract text from video and audio e-learning content.

Architecture shown on Fig. 7 contains three different processing; each one is adapted for one type of e-learning multimedia:

- Video Processing: Extract plain text from original video if it is provided by the author, otherwise apply Speech to Text API in order to extract text from video speech.
- Speech processing: Apply Speech Recognition in order to extract text from e-learning speech.
- Text processing: Extract plain text from e-learning text.

This architecture enables to have a normalized Database, which contains plain text for every e-learning multimedia sources. The next step is to extract from the plain text the important words (keywords) that will represent each multimedia content.

C. Automatic Keywords Extraction

In this subsection, we will present a keyword extraction evaluation based on the techniques described in our literature review.

1) Keyword extraction approach evaluation: In order to evaluate the performance of the different algorithms, a first approach consists of applying a manual evaluation based on human judgement to decide whether the keywords are representatives of the content of a document or not [17]. Nevertheless, manual evaluation of Automatic Keywords Extraction is difficult and time consuming.

Researchers have therefore developed some automatic evaluation systems based on partial correspondence. Automatic key phrase extraction methods have generally been evaluated based on the number of N first candidates which correspond correctly to the reference keywords. This number is then used to calculate the accuracy, recall and F-score for a set of keywords.

In the same perspective, we will compare keywords obtained by each approach with the set of Tags provided by the author of some multimedia content and which we consider to be the ground truth. So, we will use the F1 score which is based on precision and recall. Precision is defined as the number of correctly predicted keywords out of the number of all predicted keywords, and recall is defined as the number of correctly predicted keywords out of the total number of keywords in the ground truth set. Note that, to determine the correspondence between two key words, we use Porter Stemmer for the preprocessing in order to consider keywords which have the same root word.

Then, we calculated F1 for the first 20 keywords on our set of e-learning multimedia. Fig. 8 represents box plot graphs for evaluated algorithms as well as Table II presents the statistical measures (Min, Max and the mean).

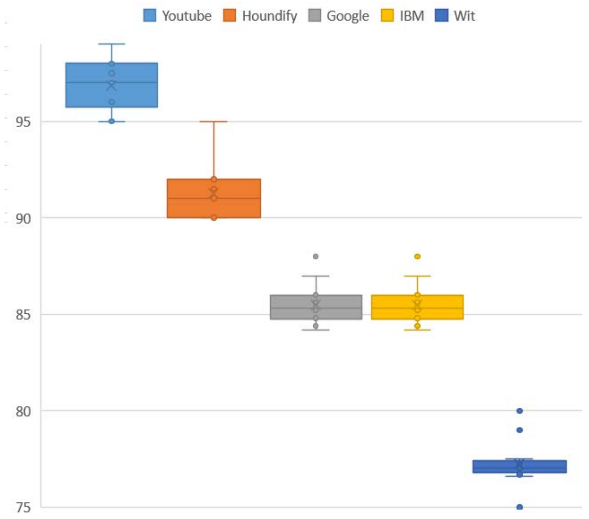


Fig. 6. WER Index of Transcription obtained from the ASR APIs.

TABLE I. WER INDEX OF TRANSCRIPTION OBTAINED FROM ASR APIS

	YouTube	Houndify	Google	IBM	Wit
WER	97,04	91,12	86,83	86,13	75,23

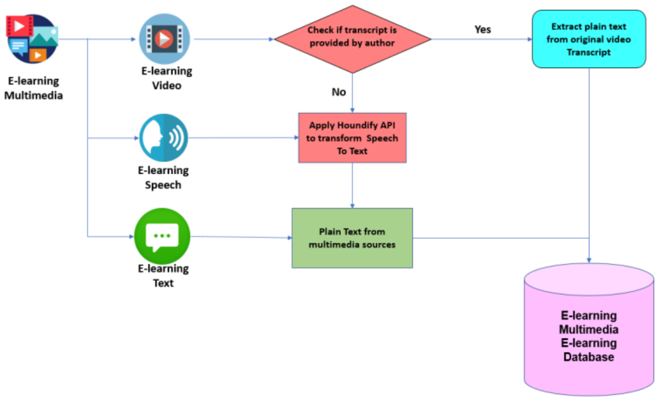

E-learning Multimedia
E-learning Video
Check if transcript is provided by author
Yes
Extract plain text from original video Transcript
No
Apply Houndify API to transform Speech To Text
E-learning Speech
Plain Text from multimedia sources
E-learning Text
E-learning Multimedia E-learning Database

Fig. 7. Architecture of the approach chosen for Multimedia Normalization (Video/Speech to Text Normalization).

Based on Fig. 8, we note that the F1-score of all approaches shows a very large variability. Thus, the relevance of the generated keywords is not stable enough. We can deduce that all approaches evaluated are not adapted to all the content of our e-learning database, and that each approach generates a set of relevant keywords only for a part of our e-learning corpus. In this context, we suggest a new framework based on ensemble methods in order to improve keywords consistency.

2) Suggested framework for automatic keyword extraction: In order to propose a framework adapted to our case study, our main idea consists of suggest an approach allowing to combine the results obtained from all approaches by using a voting system.

Basically, the first step our approach is to apply all the AKE methods in order to obtain the first N keywords with their weights according to each method.

After obtaining the keywords with their weights for each method, the second step of our approach is to normalize the weight of each keyword by dividing its weight for each method by the weight of the keyword with the maximum weight. Thus, the first keyword will have a weight equal to 1.

$$w_{ij} = \frac{w_{ij}}{\max_i(w_{ij})} \; \forall\, i \; \in \{keywords\}, \forall\, j \in \{models\}$$

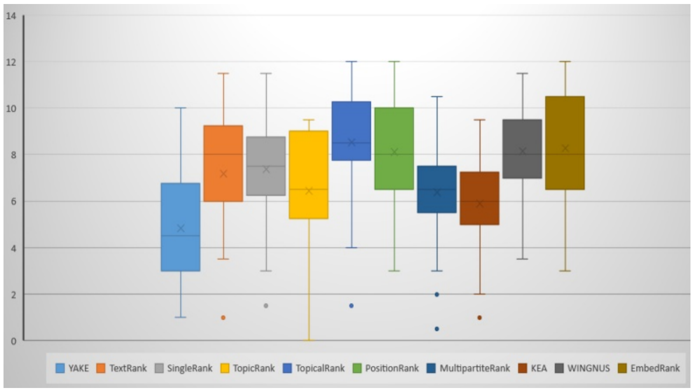


Fig. 8. Box Plot Evaluation Graph for each Keyword Extraction Approach.

TABLE II. WER INDEX OF TRANSCRIPTION OBTAINED FROM ASR APIS

	Min	**Average**	**Max**
YAKE	1	4,82	10
TextRank	1	7,18	11,5
SingleRank	1,5	7,38	11,5
TopicRank	0	6,44	9,5
TopicalRank	1,5	8,54	12
PositionRank	3	8,1	12
MltipartitieRank	0,5	6,36	10,5
KEA	1	5,88	9,5
WINGNUS	3,5	8,14	11,5
EmbedRank	3	8,26	12

Then, a global weight for each keyword is calculated by summing the normalized weights. The normalization step avoids favoring one method over another and thus allow to obtain a global weight based on a non-discriminatory vote. Also, to avoid generate similar keywords, the weights of keywords with the same root word are grouped together by considering them as a single candidate keyword.

$$w_i = \sum_{i^{ème}\, keyword\ in\ keywords\, model} w_{ij} \forall\, i \in \{keywords\}$$

Finally, candidate keywords are ranked based on their global weight and the top N are chosen as keywords to represent each e-learning multimedia content.

We notice that the choice of the models to be considered in order to build the global weight of each candidate keyword is important. Thus, to set up an adequate approach to our case study, several configurations are evaluated, namely:

- Voting system based on all methods (Keyword Vote).
- Voting system based on one method per approach (statistic, graph based, word embedding, supervised approach) (Keyword Vote 2).
- Voting system based on best methods (PositionRank, Topical Rank, EmbedRank, Wingnus) (Keyword Vote ++).

Fig. 9 represents a comparison between different configurations of our approach with the previous methods. The results obtained confirm that the proposed approach (Keyword Vote) enable to obtain results with very little variability. Thus, all the keywords generated are relevant for most of the e-learning content. In addition, Fig. 9 shows that (Keyword Vote ++) achieves the best performance. This is due to a good combination of models which stabilizes the generation of relevant keywords.

Based on the results obtained, we can deduce that Keyword Vote ++ makes allow generating a set of relevant and diversified keywords in order to represent each e-learning content. This approach is chosen to extract the keywords from the text of each multimedia content.

3) Suggested chatbot design: To design our chatbot we suggest an architecture composed with two main parts. The first part consists in designing the Chatbot Backend which contains the NLP engine allowing to understand the intention of the user and to propose the most adequate response to the needs of the learner. The second part consists in developing the Chatbot Frontend which constitutes the interface allowing the user to interact with the Chatbot. Fig. 10 shows the overall Chatbot architecture.

a) Chatbot BackEnd: The main role of the Chatbot Backend is to deal with the user's question using NLP Engine and then offer most similar answers to the learner's needs. So, to design the Chatbot Backend we use the approach proposed which is based on extracting keywords using Keyword Vote ++. Fig. 11 shows the backend architecture of the chatbot.

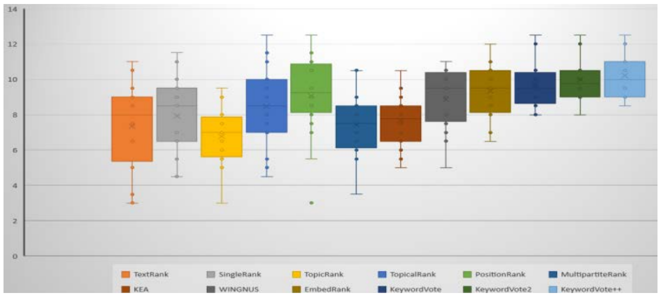


Fig. 9. Box Plot of comparison between different Configurations of our approach with the previous methods of Keyword Extraction.

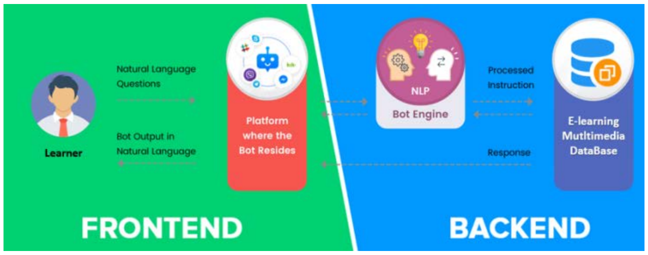

Learner
Natural Language Questions
Bot Output in Natural Language
Platform where the Bot Resides
NLP
Bot Engine
Processed Instruction
Response
E-learning Mutltimedia DataBase
FRONTEND
BACKEND

Fig. 10. Chatbot Design.

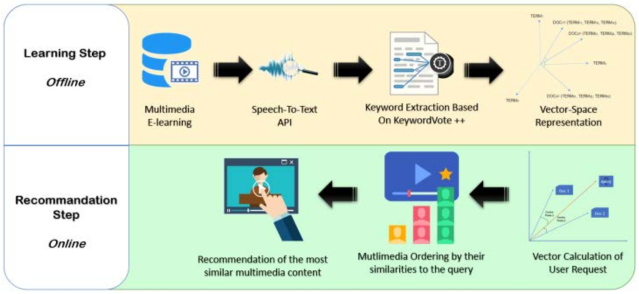

Learning Step
Offline
Multimedia E-learning
Speech-To-Text API
Keyword Extraction Based On KeywordVote ++
Vector-Space Representation
Recommandation Step
Online
Recommendation of the most similar multimedia content
Mutlimedia Ordering by their similarities to the query
Vector Calculation of User Request

Fig. 11. Backend Chatbot Architecture.

The backend process consists of two steps. In the first step, the keywords proposed approach is used to extract the relevant keywords from the e-learning content, then building the representation space of each multimedia e-learning using the terms/documents matrix. This first step is carried out in offline mode since it is not linked to user demand and allows the multimedia indexing. The second step is triggered at each user instruction and consists of calculating the representation of the query in the multimedia indexing space, then recommending to users the most similar content to his request.

b) Chatbot FrontEnd: The Chatbot Frontend consists of the interface design that allows to receive the user instructions and interact with the Backend to display the appropriate response. Thus, to ensure interaction with the user, the Chatbot must have a user-friendly interface. In fact, the Chatbot interface is based on messaging platforms to interact with the user.

In order to develop the Frontend of our Chatbot, we will use the design tools offered by messaging platforms. Our choice was towards the messenger Bot API offered by Slack [32] since on the one hand, it provides the possibility of interacting with a Python script which allows us to set up our approach developed in the Backend, and, on the other hand, Messenger is the most messaging platform used by learners to interact with each other. Thus, implementing our e-learning Chatbot on Slack Messenger has the advantage of offering Chatbot services to learners in an interface that they are used to, which will improve the user experience.

IV. Results and Simulation

A. Results

The previous chapter explained the suggested approach for designing the e-learning chatbot. Indeed, our chatbot is based on a feature space for representing multimedia content built from the most relevant keywords instead of using the overall text which is extracted from multimedia content.

In order to show the advantage of the proposed approach, we apply it on a very large corpus with predefined categories. Indeed, we will compare two approaches:

The approach which is based on the construction of the terms / documents matrix by calculating the weight of the TF-IDF for all words in document [15].

The proposed approach which is based on the Keyword Vote ++ algorithm to extract the keywords and then use the list of keywords to build the similarity matrix between the documents by calculating TF-IDF weight for just the relevant keywords.

Both approaches provide a similarity matrix which will be used to obtain a hierarchical clustering of the multimedia content. In order to validate the clusters obtained from the two approaches, we use the known categories of the corpus as a ground truth. Thus, given the knowledge of class assignments from the ground truth, it is possible to define an intuitive evaluation measure using conditional entropy analysis. Then, we measure two score that aims to identify the Homogeneity and Completeness of each clustering assignment:

- Homogeneity: mean that all members with the same cluster belong to the same class.
- Completeness: all the members of a given class are in the same cluster.

The two concepts are between 0 and 1. Thus, on the basis of its two score another measure called V-measure can be calculated. Indeed, V-measure is the harmonic mean of the two scores.

To complete our evaluation, we use the Rand Index. It measures the similarity between two partitions. Rand's index measures the percentage of correctly classified decisions and is defined as follows:

$$RI = \frac{a+b}{a+b+c+d}$$

Let X: the partition obtained from the clustering algorithm and G: the partition obtained from the ground truth. So:

- $a:$ the number of element pairs that is in the same cluster in X and the same cluster in G.
- $b:$ the number of element pairs that is in a different cluster in X and a different cluster in G.

- c: the number of element pairs that is in the same cluster in X but in different clusters in G.
- d: the number of element pairs that is in a different cluster in X but in the same cluster in G.

Fig. 12 illustrates the results obtained by applying the two approaches on our corpus.

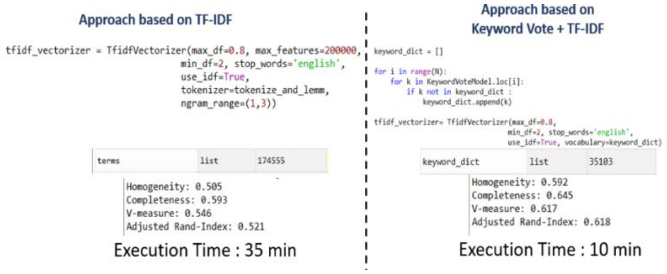
Based on the results obtained, we can confirm that the proposed approach reduces the representation space and thus makes it possible to represent documents with only 35103 terms instead of 174 555 terms (a reduction of the dimension of 80%). It also reduces the time required for the construction of the dendrogram which was reduced at 10 min instead of 35 min (a 70% decrease in execution time). Regarding the validation indices, we notice that there is a remarkable improvement in indices with an 8% gain in performance. This confirms that selecting terms based on keywords helps create create a more homogeneous tree structure for indexing multimedia content and with less execution time.

B. Simulation

In order to show the suggested chatbot in action, we propose a simulation based on different multimedia e-learning content which concerns tutorial of different Business Intelligence and Artificial Intelligence Tools. The first step is to apply our suggested Keyword Extraction approach in order to construct an indexed e-learning content. Fig. 13 show Dendrogram obtained by our approach.

The dendrogram provides a hierarchical representation of our database of e-learning contents. Indeed, we notice that multimedia contents are grouped in a way that we can easily distinguish between the different categories contained in our e-learning database. This confirms that the proposed approach is well suited to our case study. The indexed multimedia representation allows us to organize our database in order to facilitate information access and offer content adapted to each user. Indeed, this indexed database will be used into chatbot engine to response to the learner's needs. Our Chatbot design allows to interact with learners in two ways: Quick Reply and Carousel.

1) *Quick reply:* Quick reply allows to create short instant responses that can be selected by users. Indeed, we use this form to create suggestions of course categories to the user. Fig. 14 shows an example of quick reply.



Approach based on TF-IDF

```
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                   min_df=2, stop_words='english',
                                   use_idf=True,
                                   tokenizer=tokenize_and_lemm,
                                   ngram_range=(1,3))
```

terms	list	174555

```
Homogeneity: 0.505
Completeness: 0.593
V-measure: 0.546
Adjusted Rand-Index: 0.521
```

Execution Time : 35 min

Approach based on Keyword Vote + TF-IDF

```
keyword_dict = []

for i in range(N):
    for k in KeywordVoteModel.loc[i]:
        if k not in keyword_dict :
            keyword_dict.append(k)

tfidf_vectorizer= TfidfVectorizer(max_df=0.8,
                                  min_df=2, stop_words='english',
                                  use_idf=True, vocabulary=keyword_dict)
```

keyword_dict	list	35103

```
Homogeneity: 0.592
Completeness: 0.645
V-measure: 0.617
Adjusted Rand-Index: 0.618
```

Execution Time : 10 min

Fig. 12. Results comparison of the Proposed Approach with the Classic Approach for Creating a Tree Structure.

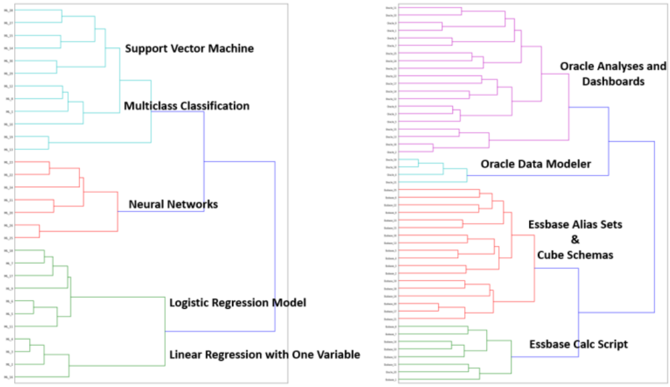
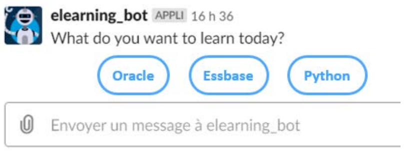

Support Vector Machine
Multiclass Classification
Neural Networks
Logistic Regression Model
Linear Regression with One Variable
Oracle Analyses and Dashboards
Oracle Data Modeler
Essbase Alias Sets & Cube Schemas
Essbase Calc Script

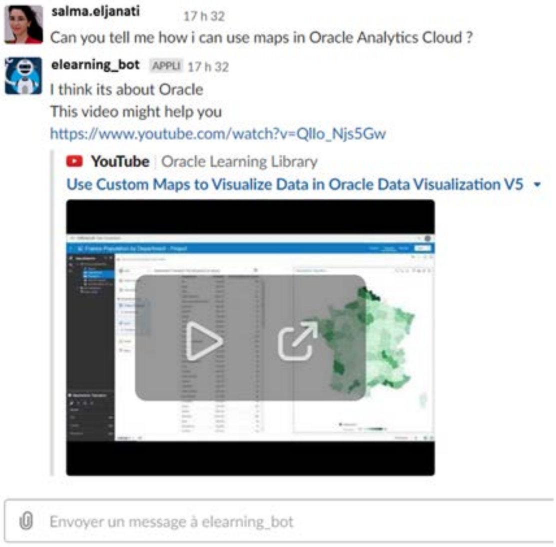
Fig. 13. Hierarchical view resulted from Multimedia Indexing.



elearning_bot APPLI 16 h 36
What do you want to learn today?

Fig. 14. Messenger Bot Quick Reply Example.

2) *Carrousel:* Another way to display results to the learner is carousel. A carousel is used when a lot of data must be presented to the learner. The buttons that accompany this form can either return a personalized message to the bot as a specialized command to trigger a flow or redirect to a URL. We use this form to recommend multimedia content which fit the user request. Fig. 15 illustrates an example of using carousel in our Chatbot.



salma.eljanati 17 h 32
Can you tell me how i can use maps in Oracle Analytics Cloud ?

elearning_bot APPLI 17 h 32
I think its about Oracle
This video might help you
https://www.youtube.com/watch?v=Qllo_Njs5Gw

YouTube | Oracle Learning Library
Use Custom Maps to Visualize Data in Oracle Data Visualization V5

Fig. 15. Carrousel example.

V. Recommendation

The proposed ChatBot design allows the learner to offer the multimedia content adapted to their needs. However, its functionality is limited and does not allow it to interact effectively in the case of general questions by the learner. One of the ways to improve ChatBot is to integrate a Chit-Chat to simulate human conversation.

Fig. 16 shows the new architecture proposed that will be the aim for our future work.

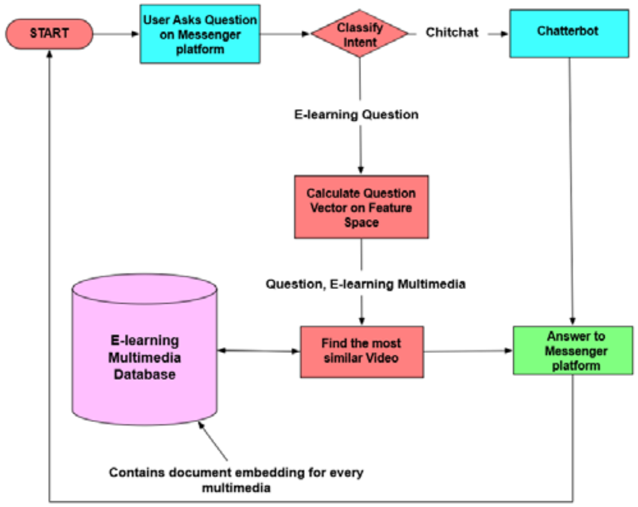

START
User Asks Question on Messenger platform
Classify Intent
Chitchat
Chatterbot
E-learning Question
Calculate Question Vector on Feature Space
Question, E-learning Multimedia
E-learning Multimedia Database
Find the most similar Video
Answer to Messenger platform
Contains document embedding for every multimedia

Fig. 16. Architecture for Integrating Chit-Chat into the Design of ChatBot.

After receiving the question from the user, a classifier will predict the class of the question and determine whether the question is related to the area of e-learning or a general question. Then, depending on the class category, the ChatBot offers an answer according to two scenarios:

- Offer the adapted e-learning multimedia using the approach proposed in this paper.
- Suggest a response from the Chit-Chat database The integration of this module makes the conversation as natural as possible.

VI. Conclusion

The aim of our study is to develop a Chatbot allowing the interaction with learners and suggest the e-learning multimedia content, which fit their learning needs. To achieve this objective, first we set up an analysis of e-learning contents in order to extract the maximum amount of information. Indeed, we propose an approach based on the Speech-To-Text APIs to extract text from different sources of multimedia.

Based on the information obtained from the extracted text, we set up the step of keywords extraction. In this step, we evaluate different algorithms proposed in the literature. Then, we conclude that they are not suitable for all multimedia contained in our e-learning database. Thus, we propose a new approach making it possible to combine the results of different approaches using a voting system. After that, we proceed to indexing of the e-learning content by constructing a tree structure allowing the organization of the information and facilitating the access to the e-learning content.

Finally, we design the ChatBot core which was divided into Backend / Frontend. On the one hand, the Backend design is mainly based on the proposed approach to indexing e-learning multimedia content and which constitutes the engine of the NLP used. On the other hand, the design of the Frontend is based on Slack Messenger platform which offers an interface facilitating interaction with learners.

Our methodology aims to have an efficient way to represent the multimedia content based on keywords. The use of keywords on our approach result on a better representation and reduce time to construct multimedia indexing. The core of our chatbot is based on this indexed multimedia content which enables it to look for the information quickly. Then our designed chatbot reduce response time and meet the learner's need.

The proposed Chatbot design allows the learner to get the multimedia adapted to their needs. However, its functionality is limited and does not allow it to interact effectively in the case of general questions by the learner. Our future work will focus on integrating a Chit-Chat to simulate a human conversation, also we will integrate voice recognition on the chatbot in order to enlarge the scope of the chatbot interactions.

References

[1] El Janati, S., Maach, A., El Ghanami, D., "Learning Analytics Framework for Adaptive E-learning System to Monitor the Learner's Activities," International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, 2019.

[2] El Janati, S., Maach, A., El Ghanami, D., "SMART education framework for adaptation content presentation". Procedia Computer Science, 2018, vol. 127, p. 436-443.

[3] El Janati, S., Maach, A., El Ghanami, D., "Context aware in adaptive ubiquitous e-learning system for adaptation presentation content". Journal of Theoretical and Applied Information Technology, 2019, 97(16), pp. 4424-4438.

[4] Yan, Z., Duan, N., Bao, J., Chen, P., Zhou, M., Li, Z., & Zhou, J., "Docchat: An information retrieval approach for chatbot engines using unstructured documents". In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016. p. 516-525.

[5] Singh, J., Joesph, M. H., & Jabbar, K. B. A., "Rule-based chabot for student enquiries". In: Journal of Physics: Conference Series. IOP Publishing, 2019. p. 012060.

[6] Zamora, J., "Rise of the chatbots: Finding a place for artificial intelligence in India and US". In: Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion. 2017. p. 109-112.

[7] Sheikh, S. A., Tiwari, V., & Singhal, S., "Generative model chatbot for Human Resource using Deep Learning". In: 2019 International Conference on Data Science and Engineering (ICDSE). IEEE, 2019. p. 126-132.

[8] Wang, Z., Wang, Z., Long, Y., Wang, J., Xu, Z., & Wang, B., "Enhancing generative conversational service agents with dialog history and external knowledge". Computer Speech & Language, 2019, vol. 54, p. 71-85.

[9] Zhang, J., Huang, H., & Gui, G., "A Chatbot Design Method Using Combined Model for Business Promotion". In: International Conference in Communications, Signal Processing, and Systems. Springer, Singapore, 2018. p. 1133-1140.

[10] Amudha, S., & Shanthi, I. E., "Phrase Based Information Retrieval Analysis in Various Search Engines Using Machine Learning Algorithms". In: Data Management, Analytics and Innovation. Springer, Singapore, 2020. p. 281-293.

[11] Koka, R. S., "Automatic Keyword Detection for Text Summarization". PhD diss., 2019.

[12] Hulth, A., & Megyesi, B. B., "A study on automatically extracted keywords in text categorization". In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006. p. 537-544.

[13] Fernandez, R. R., & Uy, C., "Keywords on Online Video-ads Marketing Campaign: A Sentiment Analysis". Review of Integrative Business and Economics Research, 2020, vol. 9, p. 99-110.

[14] Siddiqi, S., & Sharan, A., Keyword and keyphrase extraction techniques: a literature review. International Journal of Computer Applications, 2015, vol. 109, no 2.

[15] Havrlant, L., & Kreinovich, V., "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)". International Journal of General Systems, 2017, vol. 46, no 1, p. 27-36.

[16] Paukkeri M, Honkela T., "Likey: Unsupervised language independent keyphrase extraction". In: Proceedings of the 5th international workshop on semantic evaluation, Uppsala, Sweden, 2010. p. 162-165.

[17] Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A., "YAKE! collection-independent automatic keyword extractor". In: European Conference on Information Retrieval. Springer, Cham, 2018. p. 806-810.

[18] Nara, N., Sharma, P., & Kumar, P., "Page Rank Algorithm: big data analytic". 2017.

[19] Mihalcea, R., & Tarau, P., "Textrank: Bringing order into text". In: Proceedings of the 2004 conference on empirical methods in natural language processing. 2004. p. 404-411.

[20] Wan, X., & Xiao, J., "Single Document Keyphrase Extraction Using Neighborhood Knowledge". In: AAAI. 2008. p. 855-860.

[21] Liu, Z., Huang, W., Zheng, Y., & Sun, M., "Automatic keyphrase extraction via topic decomposition". In: Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010. p. 366-376.

[22] Mikolov, T., Chen, K., Corrado, G., & Dean, J., "Efficient estimation of word representations in vector space". arXiv preprint arXiv:1301.3781, 2013.

[23] Hayati, H., Chanaa, A., Idrissi, M. K., & Bennani, S., "Doc2Vec &Naïve Bayes: Learners' Cognitive Presence Assessment through Asynchronous Online Discussion TQ Transcripts". International Journal of Emerging Technologies in Learning (iJET), 2019, vol. 14, no 08, p. 70-81.

[24] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S., "Skip-thought vectors". In Advances in neural information processing systems, 2015, pp. 3294-3302.

[25] Lau, J. H., & Baldwin, T., "An empirical evaluation of doc2vec with practical insights into document embedding generation". arXiv preprint arXiv:1607.05368, 2016.

[26] Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M.,"Simple unsupervised keyphrase extraction using sentence embeddings. arXiv preprint arXiv:1801.04470, 2018.

[27] Sarosa, M., Junus, M., Hoesny, M. U., Sari, Z., & Fatnuriyah, M., "Classification Technique of Interviewer-Bot Result using Naïve Bayes and Phrase Reinforcement Algorithms". International Journal of Emerging Technologies in Learning (iJET), 2018, vol. 13, no 02, p. 33-47.

[28] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G., "Kea: Practical automated keyphrase extraction". In: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific. IGI global, 2005. p. 129-152.

[29] Nguyen, T. D., & Luong, M. T., "WINGNUS: Keyphrase extraction utilizing document logical structure". In: Proceedings of the 5th international workshop on semantic evaluation. Association for Computational Linguistics, 2010. p. 166-169.

[30] G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97, 2012. Available: 10.1109/msp.2012.2205597.

[31] Amodei, D., Ananthanarayanan, S., Anubhai, R, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin". In: International conference on machine learning. 2016. p. 173-182.

[32] Lin, B., Zagalsky, A., Storey, M. A., & Serebrenik, A.. "Why developers are slacking off: Understanding how software teams use slack". In : Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion. 2016. p. 333-336.