Auto FAQ

End-to-end Machine Learning based system for FAQ generation from paragraphs

Akash Ghosh
Computer Science and Engineering
Indian Institute of Technology
Ropar, India
2020aim1002@iitrpr.ac.in

Advisor:Dr Mukesh Saini
Computer Science and Engineering
Indian Institute of Technology
Ropar, India
mukesh@iitrpr.ac.in

***Abstract*—Auto-FAQ is an end-end automated frequently asked questions(FAQ) generator system that takes a paragraph as input and outputs the corresponding highly probable questions based on the textual content of the input. We have used cutting-edge deep Learning and Natural Language Processing(NLP) techniques for the purpose. Currently, the system is divided into three subsystems namely the Clustering Paragraph Unit, Question Generator Unit, and the Duplicate Question Detection Unit. The main purpose of the clustering unit is given a paragraph the model should be able to understand the context of it and correctly identify which cluster it should belong to, Question Generator unit takes a paragraph and generate questions from the paragraph , and the purpose of Duplicate Question Detection is to identify questions which are semantically similar so that no 2 duplicate questions are present in the final questions set.**

I. Introduction

With the growth of the internet almost all the B2B and B2C companies, government or private institutions be it educational, or healthcare have their virtual presence mostly in the form of websites or apps. According to the latest survey of 2022, there are about 22 million e-commerce sites and the numbers are increasing rapidly every day. One of the most important sections of any website is the FAQ section where the frequently asked questions around a certain product or context are curated and maintained. Traditionally FAQ pages are maintained and created by domain experts. But in this work, we are trying to automate this very task such that given the text content of a particular page we can automatically generate FAQ pages. But the complexity occurs when there are many sections of that particular web page where different web pages are built for different purposes and have different context. For example suppose we have a news website that has different sections like sports, tech, business, etc. then it will be a blunder if we put a business-related question in the sports section and vice-versa. Another issue is that FAQ questions need to be precise such that no two duplicate questions should be present. In this paper, we tried to handle these cases using cutting-edge deep learning techniques which can be efficiently used on text data.

II. Related Work

In this section, we will briefly discuss all the work that has been done in this area of FAQ generation,retrieval and classification . The early works in this direction are based on FAQ retrieval tasks from databases using different information retrieval and data mining techniques. [15] and [16] are works in this direction. A similarity with all these above papers is that they assume there is already a knowledge base for FAQ and focus is only on the retrieval and maintenance part. So, most works in this domain are in the FAQ retrieval from the database. Very less work is done in the creation and detection of FAQ questions. In [14] author uses how similar questions could be clustered using hierarchical agglomerative clustering. Though there could be cases where duplicate questions are not that significant where this approach might not be fruitful. In [3] author extracts questions from online forums and uses features like a number of follow-ups, upvotes, views, etc as features to identify FAQs.This is a very case-specific approach that is possible in online forums where such features are easily generated and utilized. The most recent work [7] is done by Jeyaraj where the idea was to generate FAQs from email data. Our work is different in the way that we are starting from the paragraph itself and not from questions. We first cluster the paragraphs based on their context and then use deep learning models like T5 [11]for neural question generation and then we remove the duplicates using sentence embedding techniques and applying transfer learning.

III. Proposed Methodology

The complete system is divided broadly into 3 subsystems. The following subsystems are discussed verbosely below:

A. Clustering Paragraph Subsystem

The task of the subsystem is that given a bunch of passages it should be able to efficiently cluster the passages based on the semantic understanding of the text. For example, suppose we have passages from three different domains like healthcare, sports, and tech, the idea is to form three mutually-exclusive clusters of passages where each passage contains questions only from a particular domain. Mathematically it can be expressed like if

$$\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_n\} \tag{1}$$

then it should be able to divide all the n paragraphs into k clusters such that

$$|\mathbf{C}_1| + |\mathbf{C}_2| + ... + |\mathbf{C}_k| = \mathbf{n} \tag{2}$$

where C_k is the k-th cluster . We use an unsupervised algorithm because often times annotation of data is a big issue. The only assumption is that we should know how many distinct clusters or domains are there in the dataset. Machine Learning algorithms cannot be directly applied to the text data. So we use various popular feature extraction techniques for converting the text into vectors of numbers. We use various NLP techniques like Tf-IDF(Term Frequency Inverse Document Frequency) [1], Sentence-Bert [13], Universal Sentence Encoder [4], and Doc2Vec [5].

For clustering the documents we use 2 algorithms namely K-means [8] and C-means [2]. K-means is a hard clustering technique where each data point belongs to a single cluster. The objective function of K means is given by:

$$\mathbf{J} = \sum_{\mathbf{j=1}}^{\mathbf{k}} \sum_{\mathbf{i=1}}^{\mathbf{n}} ||\mathbf{x}_i - \mathbf{e}_j||^2. \tag{3}$$

where k is the number of clusters n is the number of training examples x_i is the i-th example and e_j is the centroid for cluster j

On the other hand C-means is a soft-clustering technique where each datapoint may belong to more than one cluster.It generally give better performance when data is more overlapped.The Objective function is given below:

$$\mathbf{J}_i = \sum_{\mathbf{j=1}}^{\mathbf{k}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mu_{ij}^m ||\mathbf{x}_i - \mathbf{e}_j||^2. \tag{4}$$

where μ_{ij}^m is the degree of membership of x_i in cluster j. But as these clustering techniques works on distance based metrics they sometimes suffer when data has high dimensions.That's why we use PCA to reduce dimensions and then apply clustering algorithms on the top of it.We have tried our experiments on variance ratio of 95% ,90%and 85% .In the figure 1 its shown how cumulative varience score varies with the number of dimensions in case of sentence bert.Sentence bert converts every sentence to 768 dimensions as we keep decresing the dimensions the varience score also decreases.With 145 dimensions we get around 95% varience ie 95% of the information is retained with 145 dimensions.Similarly we are getting 90% and 85% varience with 85 and 65 dimesnions respectively.

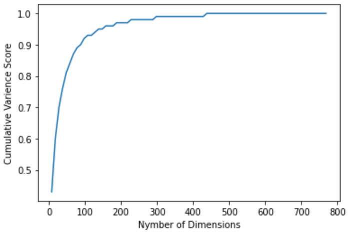


Fig. 1: Number of Dimensions VS Cumulative Varience Score

B. Question Generator Subsystem

The task of this subsystem is to generate answer agnostic questions from context paragraph.We use T5 model(Text to Text Transfer Transformer Model) [11] for this task.T5 treats every problem as a sequence to sequence problem.The primary architecture of T5 is simple encoder-decoder model of the transformer.For encoder it uses Bert model and for decoder part it uses Generative Pretrained Transformer(GPT)-2 model.Here we trained the T5 model on the famous Stanford Question Answering(SQUAD) dataset [12].In SQUAD dataset we have context paragraphs followed by bunch of corresponding questions and answers.This dataset was originally designed for the purpose of training question-answering models but we can easily transform the dataset for our task.We have formatted the dataset in 3 different ways for our neural question generation task.In first case it was all questions in a single line format.In this case, a single example consists of context followed by all the questions in a single line separated by special characters like [SEP].For the second and third case we use single question per line format where every context paragraph is followed by a single question.If a context has n questions then that same context is duplicated n times such that each training example will contain context followed by a single question.The different data representation pipeline is shown in the figure-2.

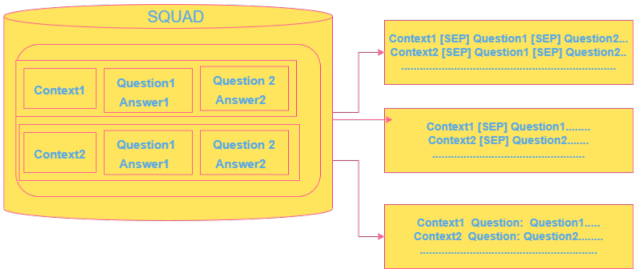


Fig. 2: Data Preparation Pipeline for Question generation

C. Duplicate Question Detection Subsystem:

One of the issue with the questions generated from subsystem-2 is that there are many questions which are semantically similar i.e there are duplicate questions.For example-”What is a good programming language to learn for an intermediate programmer?”,”What's the best language for an intermediate programmer to learn?” are duplicate question pairs.So,we have to remove those duplicate questions to generate the final FAQ set.For this task,we use sentence bert which is a modification of the pretrained bert networks .Sentence bert uses siamese and triplet network structure to create embeddings of the sentences which then can be compared using various distance measures like Cosine-distance or Manhattan-distance similarity to check how those sentences are semantically similar.We consider cosine similarity for this task.One of the major challenges of that task is choosing the similarity threshold value as the cosine similarity ranges from -1(highly disimilar) to 1(highly similar).For this we use the

TABLE I: Results for K-Means clustering

Vectorization-Technique	**Unprocessed Text**		**Processed Text Varience-100**		**Processed Text Varience-95**		**Processed Text Varience-90**		**Processed Text Varience-85**	
–	**Accuracy**	**Silhoutte-Score**	**Accuracy**	**Silhoutte Score**	**Accuracy**	**Silhoutte Score**	**Accuracy**	**Silhoutte Score**	**Accuracy**	**Silhoutte Score**
Tf-Idf	69.32	0.002	89.32	0.014	86.10	0.029	85.9	0.030	85.57	0.032
Sentence-Bert	47.58	0.071	70	0.070	71.14	0.074	70.4	0.074	70.20	0.084
Universal Sentence Encoder	**95.83**	**0.083**	**74.2**	**0.069**	**74.2**	**0.072**	**74.2**	**0.072**	**74.2**	**0.081**
Doc2Vec	6.77	0.00	10.20	0.00	6.17	0.001	6.51	0.001	5.63	0.001

TABLE II: Results for C-Means clustering

Vectorization-Technique	**Unprocessed Text**		**Processed Text Varience-100**		**Processed Text Varience-95**		**Processed Text Varience-90**		**Processed Text Varience-85**	
–	**Accuracy**	**Silhoutte-Score**	**Accuracy**	**Silhoutte Score**	**Accuracy**	**Silhoutte-Score**	**Accuracy**	**Silhoutte-Score**	**Accuracy**	**Silhoutte-Score**
Tf-Idf	38.18	-0.004	44.83	0.004	37.9	0.010	37.9	0.010	37.9	0.010
Sentence-Bert	23.55	-0.020	36.37	0.043	36.37	0.043	36.37	0.043	36.37	0.043
Universal Sentence Encoder	32.68	0.030	27.04	0.017	27.04	0.017	27.04	0.017	27.04	0.017
Doc2Vec	15.43	0.001	14.76	0.001	20	0.001	19.86	0.001	19.86	0.001

TABLE III: Results for T5 Model Finetuning scores for Question Generation

Format	BLEU-4	ROUGE-L	METEOR SCORE
Context Followed by many Questions seperated by [SEP]	0.144	0.245	0.305
Context Followed by Single Questions seperated by ”Question:”	0.133	0.252	0.355
Context Followed by Single Questions seperated by [SEP]	0.132	0.253	0.354

”Quora Question Pairs Dataset” .It has around 4 lakh pairs of questions with labelled output is_duplicate,if is_duplicate=1,it means they are similar and 0 for vice-versa.We iterate the threshold values from 0.40 to 1 and incrementing each iteration by 0.1 and use precision,recall and F1 scores for finding the optimal threshold value.That threshold value could be decided to check if a particular question pair is duplicate or not.The mathematical representation for this is shown below:

$$f(q1,q2) = \begin{cases} 1, & \text{if } cosine_similarity(q1,q2)) \geq \textbf{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Here q1,q2 represents the sentence embedding of question1 and question2 respectively. The complete Architecture of the proposed system is shown in Fig-3

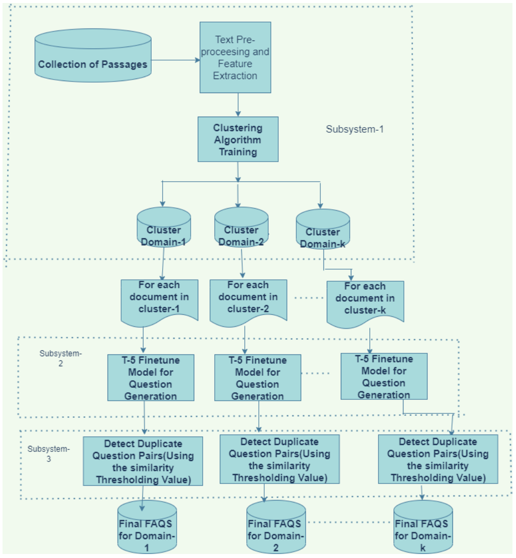
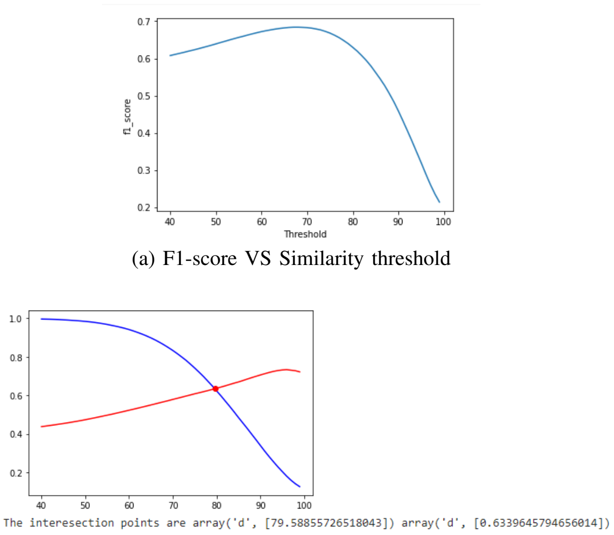

Collection of Passages
Text Pre-proceesing and Feature Extraction
Clustering Algorithm Training
Subsystem-1
Cluster Domain-1
Cluster Domain-2
Cluster Domain-k
For each document in cluster-1
For each document in cluster-2
For each document in cluster-k
Subsystem-2
T-5 Finetune Model for Question Generation
T-5 Finetune Model for Question Generation
T-5 Finetune Model for Question Generation
Subsystem-3
Detect Duplicate Question Pairs(Using the similarity Thresholding Value)
Detect Duplicate Question Pairs(Using the similarity Thresholding Value)
Detect Duplicate Question Pairs(Using the similarity Thresholding Value)
Final FAQS for Domain-1
Final FAQS for Domain-2
Final FAQS for Domain-k

Fig. 3: Architecture of the proposed System

IV. Results and Discussion

There were three different subsystems which work independently to achieve the single goal.So the results of each subsystem will be discussed separately.The main dataset used for this task is the ‘BBC News Dataset’.The data set has around 1500 paragraphs from the domain namely Tech,sports,business,politics and entertainment.There were 346 paragraphs on sports ,336 paragraphs on business ,274 paragraphs on politics,273 paragraphs on entertainment and 261 tech paragraphs.The average number of words in a particular paragraph is 415,median is 361 and maximum is 3519.So,the task of subsystem-1 is to cluster the documents into 5 distinct groups based on the domain knowledge of the data.The experiments were performed both in processed and unprocessed text.For text processing and cleaning we remove stop words,punctuation,digits,apply contractions and lemmatisation.The metric used for this purpose is accuracy and silhouette score.Results of K-means clustering and C-means clustering are given in Table I and II respectively. From the



(a) F1-score VS Similarity threshold

```
The interesection points are array('d', [79.58855726518043]) array('d', [0.6339645794656014])
```

(b) Precision(Orange)-Recall(Blue) Values VS Threshold

Fig. 4: Performance metrics evaluation for Duplicate Question Subsystem

results we can conclude that K-means performed better for the given data.The reason possible is that the data is not much overlapped and every data point here belongs to one cluster center but might not be the case every time.In cases where we get overlapped data,C-means performance will be better.Here the performance of K means coupled with Universal Sentence Encoder worked best and gave an accuracy of 95.83 and silhouette score of 0.083 on unprocessed text.

For Question Generation System the metrics used are BLEU-4 [10],ROUGE-L [9] and METEOR [6].From table III,we can see the scores of model-2 and model-3 are very close.The reason could be that the data format in the cases was same(context followed by single question).One observation we found that in both case-2 and 3 we have to increase the number of returning sequences parameter to get non-duplicate questions while in case-1,number of duplicate question generation is comparatively low. For Duplicate Question Detection subsystem we use sentence-Bert for getting the embedding and use cosine similarity to check if they are duplicate or not.Now idea is to find the similarity threshold value.A low threshold may results in high recall and low precision and vice versa.From figure-4a we can see that the highest f1-score was around the threshold value 77 which is 0.77 in threshold terms.From figure-4b,we find the intersecting point of the precision and recall values.Here we find the intersecting value was 79.58 i.e 0.79 in threshold terms.So approximately we can say the optimal threshold value to decide whether a decide is duplicate or not is somewhere between 0.77 and 0.79.So,in the final experiments we choose a threshold value of 0.78, i,e,the questions are declared duplicates if the cosine-similarity value is greater than 0.78.But we have to check other techniques too as the highest f1-score using the sentence-Bert model is around 0.68.

V. Conclusion and future-work

In this paper we discussed how FAQs could be generated from a bunch of paragraphs which belongs to some domain using Machine Learning techniques.This system could also highly beneficial for all business enterprises and government institutions.Though the work has some major limitations.The ‘BBC Dataset’ is a relatively smaller dataset and so we cannot say similar results will follow for large datasets too,because if the clustering technique does not work well for larger datasets the performance of the overall system will get impacted a lot.Secondly the system does not check the grammatical correctness of the questions which are generated which is generally expected in a FAQ forum.Also,when we find semantically similar questions we are just randomly selecting one of them,but this should not be the case,we should select the best question which could represent all those bunch of duplicate questions,so this issue also needs to be handled.

References

[1] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[2] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

[3] Ankita Bihani, Jeffrey D Ullman, and Andreas Paepcke. Faqtor: Automatic faq generation using online forums. Technical report, Stanford InfoLab, 2018.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[5] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[6] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[7] Shiney Jeyaraj and T Raghuveera. A deep learning based end-to-end system (f-gen) for automated email faq generation. *Expert Systems with Applications*, 187:115896, 2022.

[8] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[9] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[14] Renuka Sindhgatta, Smit Marvaniya, Tejas I Dhamecha, and Bikram Sengupta. Inferring frequently asked questions from student question answering forums. *International Educational Data Mining Society*, 2017.

[15] Eriks Sneiders. Automated faq answering: Continued experience with shallow language understanding. In *Question Answering Systems. Papers from the 1999 AAAI Fall Symposium*, pages 97–107, 1999.

[16] Steven D Whitehead. Auto-faq: An experiment in cyberspace leveraging. *Computer Networks and ISDN Systems*, 28(1-2):137–146, 1995.