Intelligent FAQ Search (IFS)

Kunal Joshi [Department of Computer Science, University Of Mississippi]

Guided By: Dr. Dawn Wilkins

***Abstract— IFS* is a search mechanism that gives smarter and better result than regular keyword search. It uses natural language processing to search more efficiently through the FAQs. The main idea behind this project is to implement a search mechanism which gives user more desired output by recognizing what they actually want. It uses various strategies like n-grams, synonyms, word count, noun-phrases, etc. to find a better and more suitable solution.**

Keywords—natural language, intelligent search, NLP, NLTK, smart search, smart FAQ search

I. INTRODUCTION

Frequently Asked Questions (FAQs) are a list of questions and answers relating to a particular subject, especially one giving basic information for users of a website. Also known as *Questions and Answers (Q&A)*, these are listed questions and answers, all supposed to be commonly asked in some context, and pertaining to a particular topic. The format is commonly used on email mailing lists and other online forums, where certain common questions tend to recur. In the case where the FAQ list is too long, we are provided with simple keyword search for searching. So, we introduce Intelligent FAQ Search (IFS) a smarter way to look through the FAQs which will give more precise search results using Natural Language Processing (NLP) with Python. We have also developed a modified version of keyword search specially for FAQ search.

II. M-KEYWORD SEARCH

A *keyword search* looks for words anywhere in the text file or in a given record. Keyword searches are usually used for a subject search when you do not know the standard subject heading or in some cases when we have incomplete information about something. It returns all the matching instances of the words searched as the output.

M-Keyword Search is a modified version of the generic keyword search that we developed for searching FAQs. Here, instead of looking through the whole document it looks for keywords within an individual FAQ. It maintains a count of the keywords that were found in each FAQ. It also keeps track of the number of instances of each keyword in each FAQ. When the model is done searching, it evaluates the results and the FAQ with highest number of keywords match is considered to be the best one. But it often happens that there are number of FAQs that has all the keywords matched. In that case the total number of occurrence of the matched keywords come into play. In such cases, the FAQs that has maximum keywords matched and also has the highest total keywords occurrences is considered to be the best and hence becomes the output of this model. Although, it is possible that after all this, we can get more than one FAQ as output.

III. IFS

Natural-language processing (NLP) ([1], 2018) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process natural language data. In laymen terms, it focuses on making a computer learn, understand and speak the human language. There are three main challenges in NLP – Speech Recognition, Natural-Language Understanding and Natural-Language Generation. The challenge on which we are focusing here is natural-language understanding. Our goal here is to try to make computer understand what the user is looking for and then find it for them; unlike simple keyword search which blindly searches without any understanding.

In theory, since the new model that we developed tries to understand the search query, it should be better and smarter than M-Keyword search. Also, the new model is designed in a way that we get the most suitable answer according to the query. It is aimed to be more accurate and more intelligent than the good old generic search. Hence, the name ***Intelligent FAQ Search (IFS).***

IFS uses ***Natural Language Tool Kit (NLTK)*** (Steven Bird E. L., 2004) to try to overcome the challenge of natural-language understanding. *NLTK* (NLTK, 2017) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. It has been called "a wonderful tool for teaching, and working in, computational linguistics using Python" and "an amazing library to play with natural language."

NLTK has to be downloaded and imported in the program in order to use its various packages. There are many amazing and useful approaches in *NLTK* that are helpful to make the human language more understandable for the computer. *IFS* takes advantage of few of those approaches which makes it intelligent. Although for this project not all of those are applicable or in some cases not helpful. *IFS* uses three of those approaches – N-grams, NP-chunking, Synonym Search. The way this model works is that all these approaches are implemented to each and every FAQ with reference to the query-string. Query string is the query user will use to search. For determining the better match, we keep a scoring system. With the all the approaches that are being used, scores are incremented for each of them when a match is found. The priority among the three approaches is set through these scores. Setting the priority with the scores is tricky because in some cases the scores from different approaches overlap each other which leads to wrong output. After searching through the file, IFS gives the FAQ/s with the best score as the search output.

IV. Strategies used in IFS

A. N-grams Approach:

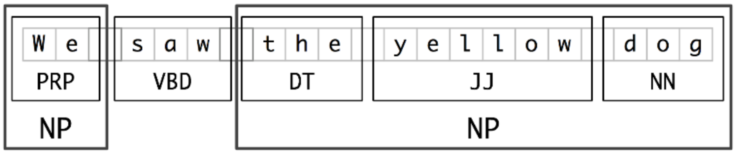
An ***n-gram*** is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called *shingles.* For making an n-gram packet, every word in a sentence or every syllable in speech is treated as a gram and the packets are formed according to the values of n. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram". English cardinal numbers are sometimes used, e.g., "four-gram", "five-gram", and so on.

For example, the bigram of a sentence: "*How can I apply?"* will be [*How can*, *can I*, *I apply?*]. For this project, the sentence will be the query-string used for searching. The n-gram packets used here are trigrams. Once the trigrams are generated, IFS searches for the trigrams in each and every FAQ for an exact match. For every match found, the score of the respecting FAQ is raised by 20.

For instance, query-string is "How do I pay my bills online". The trigrams generated for it will be ['How do I', 'do I pay', 'I pay my ', 'pay my bills', 'my bills online'] i.e. 5 packets. If a FAQ matches with 2 of these packets, the score of that FAQ will increment by 40. After searching for the trigrams, IFS takes the next approach.

B. **Noun-Phrase Chunking** (Steven Bird E. K., 2009, p. Ch. 7)

Noun phrase chunking or ***NP-chunking*** is where we search for chunks corresponding to individual noun phrases. NP-chunks are often smaller pieces than complete noun phrases. One of the motivations for this difference is that NP-chunks are defined so as not to contain other NP-chunks. Consequently, any prepositional phrases or subordinate clauses that modify a nominal will not be included in the corresponding NP-chunk, since they almost certainly contain further noun phrases.



In the example above, the letters bellow each word are called the PoS tags. PoS tags are always abbreviated. For the above example PRP stands for personal pronoun, VBD is verb, DT is determiner, JJ stands for adjective and NN is noun. The good thing about NP-chunking is that the user gets to decide the grammar for the chunks. The grammar is defined as regex of the position tags. In this project grammar is defined as: ”**NP: {<DT>?<JJ>*<NN>*<NNS>*}”**. It means an NP-chunk is a chunk were a single determiner (DT) is followed by zero or more adjectives (JJ) which are followed by zero or more singular nouns (NN) which is again followed by plural nouns if any.

This strategy forms a chunk/s from the query-string which are then looked up in each and every FAQ for a match and just like n-grams, if the chunk is present in a FAQ, the score is further incremented by 10 for every chunk found. The first two approaches works mostly all the time, but in some cases if IFS is not able to find a suitable match, it takes the third approach.

C. Synonyms Search (WordNet, 2017)

It is mostly referred as ***Synsets***. A *Synset* is a set of synonyms that share a common meaning. Each synset contains one or more lemmas, which represent a specific sense of a specific word. Synset uses *WordNet* library which is a large lexical database of English that groups nouns, adjectives, verbs which are similar and calls them synsets or synonyms. Synset creates a list of synonyms for a word with similar meanings.

This approach helps the model when the users don’t use the exact words as in FAQs. Synset tries to identify what the user wants that is close to what is already in the FAQ. This strategy is more helpful when the first two strategies fail to find better result. While using this approach, IFS creates 3 synonyms of verbs, adjectives and nouns from the query-string. It checks for the new phrases by replacing them with the actual words. If it finds a match in a FAQ, same thing – the score gets incremented by 10.

This approach sometimes overlaps the scores of the other two approaches resulting in wrong output. That is the reason that this approach is used only as a contingency.

V. PERFORMANCE EVALUATION

For evaluating both the models, we used three different text files with different number of FAQs in each. The three tables below show the results of the files with 24, 43 and 130 FAQs in that order. First column shows the keywords used for searching with M-Keyword model along with the query which is used to search in IFS model. The second column shows the correct/expected FAQ by the user. The third and fourth columns are the output FAQs by both the models. The reason behind the models give better output than the other is mentioned in the row below their respective results. The output from both the model is in bold if the output is same as the expected output.

TABLE 1

Sr. No.	Keyword/Query	Expected FAQ	M-Keyword	IFS
1.	Keywords: profile logging problems Query: I am having trouble logging into my profile	Q5	**Q5**	**Q5**
2.	Keywords: Confirmation for submit application. Query: Do I get a confirmation after submitting my application	Q23. After I submit my application, will I receive a confirmation?	Q1 Q4 Q20 Q21	**Q23**
	IFS do better because of the NP-chunk approach. It found a match for “application” and “a confirmation”			

3.	Keywords: after registering Query: what happens after completing my registration	Q7	**Q7** Q8 Q23	**Q7**
4.	Keywords: reset password Query: What should I do to reset my password	Q11. I have forgotten my password. What should I do?	Q9 Q10	Q10 **Q11**
	IFS found the correct one because of the Trigram Approach. It found a match for “What should I” and “should I do”			
5.	Keywords: my information not in the options Query: what if my information is not in the given options	Q16. What should I do if my relevant information is not included in one of the drop-down options?	Q5 **Q16**	**Q16**
6.	Keywords: offline application Query: can I make an offline application	Q3	**Q3**	**Q3**

TABLE 2

Sr. No.	Keyword/Query	Expected Answer	M-Keyword	IFS
1.	Keywords: graduate assistant qualifications Query: what are the qualification for being a graduate assistant	Q33	Q30 **Q33**	**Q33**
2.	Keywords: remove Graduate Dean's hold Query: what should I do to remove graduate dean's hold	Q27	**Q27**	**Q27**
3.	Keywords: students enrolled in graduate school university of Mississippi Query: how many students enroll in graduate school of university of Mississippi	Q41. How many students attend the University of Mississippi Graduate School?	Q5 Q21	**Q41**
	IFS gets it correct because of the Trigrams Approach for matching "how many students" and "University of Mississippi"			
4.	Keyword: Candidacy Query: What is candidacy	Q22	Q21 **Q22**	Q21 **Q22**
5.	Keywords: apply in multiple graduate programs Query: how can I apply in	Q15. Can I apply to more than one graduate program?	Q8	Q7 **Q15**

	multiple graduate programs			
	Trigram approach make IFS to get the correct FAQ with “can I apply”			
6.	Keywords: minimum credit hours for each semester Query: what are minimum credit hours for each semester	Q19. Are there minimum and maximum enrollment requirements per semester?	**Q19**	Q18
	IFS gives wrong output because the score of Trigram match “for each semester ”(20) buries the score of NP-chunk “minimum” (10)			
7.	Keywords: check application status Query: how do I know my application status	Q6	**Q6**	**Q6**
8.	Keywords: get scholarship Query: How do I get scholarship	Q34	**Q34**	**Q34**
9.	Keywords: travel assistance Query: Can I get travel assistance	Q38. Does the Graduate School provide assistance for travel?	**Q38**	**-**
	IFS was not able to find any match with any of the three approaches			
10.	Keywords: eligible committee members Query: Who can be on my committee	Q30	**Q30**	**Q30**

TABLE 3

Sr. No.	Keyword/Query	Expected Answer	M-Keyword	IFS
1.	Keywords: get life insurance quote Query: How can I get life insurance quote	Q65	**Q65** Q67	**Q65**
2.	Keywords: pay new bills Query: how can I pay my new bills	Q19. How will I know when I have new bills to pay?	**Q19**	Q43.
	Q43. How can I pay for my insurance if I decide to purchase it online today? The score of Trigram "How can I" in IFS overlaps the NP-chunk score of "new bills"			
3.	Keywords: make online payment Query: How do I make online payment	Q23	**Q23**	**Q23** Q121 Q123
4.	Keywords: State Farm's routing number Query: What is State Farm's routing number	Q88. What is State Farm Bank's routing number?	Q1 Q3 Q89	**Q88**
	Trigram match for "What is State" and the NP-chunk "routing number"			
5.	Keywords: payment due date Query: How can I know my payment due date	Q106	Q22 **Q106**	**Q106**
6.	Keywords: buy policy online	Q59. Can I purchase my	Q5	…

	Query: Can I buy policy online	policy online?	Q10 Q20	**Q59** …
	IFS finds a match using synsets “buy” = “purchase”			
7.	Keywords: create personal inventory Query: Can I create my personal inventory	Q58. How do I create an inventory of my personal property?	**Q58**	-
	IFS couldn’t find any match using either approaches			
8.	Keywords: lock my interest rates Query: How do I lock my interest rates	Q103	Q95 **Q103**	**Q103**
9.	Keywords: Solicitation Notice Query: Why am I getting solicitation notice	Q81	**Q81**	Q5 **Q81**
10.	Keywords: Register for online account Query: How can I register for online account	Q37. How do I register for an online account?	Q5 Q20	**Q37**
	Trigrams match “I register for”			

From the above three tables, it is clear that IFS has more correctness than M-Keyword most of the time but even when both are correct, IFS gives precise output while M-Keyword has the expected output way below in the list. It makes it much easier for the user when what they want is right there in front of them instead of looking for the right thing even after a search. In rare cases, even the IFS messes up and gives wrong output. Mostly, the reason for this is the overlap of scores from different approaches due to which the scores of other FAQs are increased while the correct answer gets buried with a lower score. Also, it can be seen that in very rare case, IFS is not able to find any suitable answer to the query. The conclusion to that problem is that it doesn't have enough information on how to search in other ways. After comparing the results from both the models, the results conclusion is as below:

File [# of FAQs]	**M-Keyword**		**IFS**	
	Correctness	Vs. IFS	Correctness	Vs. M-keyword
Faq.txt [24]	90%	Almost never	90%	25%
UM.txt [43]	Approx. 80%	5%	Approx. 85%	25%
Insurance [130]	Approx. 65%	5%	Approx. 75%	35%

Looking at the overall results, it is established that IFS works better than M-Keyword search almost all the time, even when both are correct. It can also be said that as the number of FAQs go higher, the correctness reduces for both the models. On the bright side, IFS gets more advantage over M-keyword model as the file gets bigger.

VI. CONCLUSION

Looking at the results that we got, it is safe to say that IFS has more correctness rate than M-Keyword Search. Even in the cases where both models are correct, IFS has advantage in giving precisely correct output than M-keyword search. As a practical result, user doesn't have to go looking for the correct FAQ even after searching for one. The reason behind that is IFS tries to collectively understand what user wants to find.

Although there are some cases where IFS couldn't find what the user is looking for and gives a completely incorrect output or in rare cases no output at all. This is because, it doesn't have enough understanding of what the user is looking for. For that, more approaches needed to be implemented along with the ones in IFS.

VII. FUTURE WORK

The proof of IFS giving wrong output shows that there is more room for improvement in the IFS model to make it more natural language understandable. More approaches can be implemented using NLP that can make it more correct, efficient and accurate. One of the approaches could include something that makes it understand different types of sentence formations. With some modifications, IFS could be implemented to larger files or datasets like forums.

Also, feedback system can be used with it for using the data-mining approach. So, the next time when different users has a similar query, it can take into consideration the feedback of the previous searches which can help in prioritizing the output FAQs.

VIII. Bibliography

[1]. (2018, April 24). *Natural Languare Processing.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Natural-language_processing

NLTK. (2017, September 24). *NLTK 3.2.5 documentation*. Retrieved from Natural Language Toolkit: https://www.nltk.org/

Steven Bird, E. K. (2009). *Natural Language Processing with Python.* O'Reilly Media.

Steven Bird, E. L. (2004). NLTK: the natural language toolkit. *ACLdemo '04 Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (p. 4). Stroudsburg, PA: Association for Computational Linguistics Stroudsburg, PA, USA ©2004.

Wiki. (2018, April 24). *Natural Language Processing*. Retrieved April 30, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Natural-language_processing