Using Archived Comments on Learning Videos as a Resource for Question Answering

Md Masum Billah, Prof. Dr. H. Ulrich Hoppe, Dr. Tobias Hecking

University of Duisburg-Essen, Germany
billah.md-masum@stud.uni-due.de
hoppe@collide.info
hecking@collide.info
Department of Computer Engineering (ISE)

DOI: 10.29322/IJSRP.8.12.2018.p8459
http://dx.doi.org/10.29322/IJSRP.8.12.2018.p8459

Abstract

In video-based tutoring systems, users can ask the questions and receive the answers from the system and aim at facilitating a learner in video-based learning. i.e., while watching a tutorial video, the users can enter a question (or some keywords) and retrieve potential helpful comments from the Q/A threads of Khan Academy tutorial videos using the textual overlap between the query and possible answers. It helps to find comments answer regarding tutoring systems. Moreover, this system itself is related to tutoring systems. It can be used to query Khan Academy comments according to some given keywords.

Keywords: Tutoring Systems, Q/A Systems, Python, Web development Tools: JavaScript

1. Introduction

Nowadays people can learn several concepts from the web-based learning platform. Online learning is the prevalent way of learning but when a learner wants to know the video topics discussion follow comments by other learners. For instances, when a user/ learner wants to know about the video comments on the topic which has been written by existing users or learners. He/she might knows other peoples discussion about the topics. This paper, the approach to developing tutorial components based on video targeted comments. It takes some sophisticated knowledge-based content technology to store structured and unstructured information.
The ultimate objective is text mining perception is to collect the useful information. Our goal is to retrieve question to answer pairs for videos as well as automatically. The availability of active discussions in this site provides plentiful opportunities for knowledge summarisation for searching a query and retrieve the relevant answers.
In this paper, we have presented an approach to web-based learning purposes to implement learning experience to build by some videos and text.
Web-based learning meant to used several videos, and it includes various features for education, practicing, progress and teaching tools (Sampson et al., 2014). TREC-8 was the first large-scale evaluation of domain-independent question answering systems (Voorhees and Tice, 2000).

In this work focuses on extracting comments on the selected online video learning. Content analysis techniques used to derive the structured and describe the conceptual relations (Daems et al., 2014). In this paper, to provide the user's comments on e-learning video. Our approach could be retrieved a similar question and answer pairs.

In this work, use the freely available resource of the question and answer sets to define intelligent tutoring systems(Khan Academy) which perceive the user's corresponding questions and answers. Data consists are set of question and answer threads so that we know data retrieve and extraction, text processing, and network analysis.

2. Related works and background

The EU Juxtalearn project (Daems et al., 2014) developed the science concepts to the create and share the videos to specify the targeted knowledge. These are the relevant concepts of the domain. Used the content analysis techniques to extract the learner's idea to declare in textual artifacts. The result of the analysis perceived student concepts and other ontology techniques suggest as "problems of understanding" and "misconceptions" (Daems et al., 2014). These are evaluated to find the automated analysis of the textual artifacts.

The ongoing web-based learning systems provide the user comments that hold to understand the video concepts. This system is capable of interacting with the tutoring components and users. We search the query into a set of categories regarding the domain concepts (Daems et al., 2014). Ontology is the concepts of domain and relationship that exists among the domain ontology.

2.1 Intelligent Tutoring System (ITS)

So far, a limited amount of research has been done on web-based learning compared to the intelligent tutoring systems. In our thesis, Intelligent tutoring system has been used for interact directly to combine the students and tutors. Any other analytical techniques for existing interfaces can also be applied. The intelligent tutoring systems have been used to prepare learners into beforementioned different domains. We have used intelligent tutoring systems to expand the variety of instructional functions such as subjects domain, and students can handle the system response.

The domain model to solve the complicated problems where the student can practice and have new discussions. For tutoring model, distinctive ways of learning on a particular topic. Different domain reused by the learner.

Intelligent tutoring systems used artificial intelligence techniques to supports "Learning by doing ("H. Ulrich Hoppe") with computer-based systems, specifically help to learners problem understanding the mixed-initiative approach which can ask questions.

2.2 Text Mining

Text mining applies in the various learning process. It takes some complex knowledge-based content technology to store structured and unstructured information. The ultimate objective is text mining perception is to collect the useful information. To develop the text mining initiated by text documents (e.g. book in a library). But now text mining development moved to extract the textual information using natural language processing techniques is the comprehensive techniques are then used to extract relevant information. Text mining techniques, which can develop and enhance (Miner, Elder IV, and Hill, 2012) the artificial intelligence.

Text mining is used for knowledge corpus, extract the text information, classifying, clustering to meaningful understand (Gottipati, Lo, and Jiang, 2011). In the automatic extract, the text document of information (Krallinger and Valencia, 2005) to improve the database annotations with the knowledge base.

Text mining that can form text collections including retrieving and classified to combined user queries, extensive analysis to obtain specific topics, these are called text mining. Text mining tools used to develop the unknown training dataset from specified resources (Gupta and Lehal, 2009).

Text mining can represent by compliant ways of information management, analysis, and interpretation. Thus text mining can develop the fists of data mining to the ability to deal with textual materials. The purpose of text mining a field of various data mining (Navathe and Ramez, 2000) to find the model from the corpus. In text analysis, design the structured and semi-structured full-text documents.

2.3 N-Gram Language Model

This model could predict also assign to the likelihood of an entire sentence. N-gram model is the contiguous sequence of n words from given a set of sample text to estimate the probability (Jurafsky and Martin, 2009) and (Chen and Goodman, 1999).

Our approach to solving the predicting a word from the previous word in a sample of text. In particular, we consider n-gram models based on the set of words. We also discuss several analytical concepts for specifying words to categories based on the frequency in section 2.6 of their co-occurrence with other words (Brown et al., 1992).

The performance of a language model component is evaluated by the perplexity, which measures both the task complexity and the adequacy of the language model. Once the acoustic and language models are determined from a training corpus, the recognition process consists of maximizing the product which is performed through the search techniques.
The probability of each word does not depend on all the previous words in the sentence, but only on the N-1 previous ones. Therefore, the N-gram probability assigned to a sequence of words W is:

$$P(W) = \prod_{x=1}^{m} P(w_k \mid w_1^{k-1})$$

Where W = w1 . . . wm, and wk−1 represents the string wk. we discuss the language model which can predict the probabilities of sentences and sequences of words using the following cases:

N-gram is a sequence of N words like "could you explain what is Programming Language?”

Bi-gram is a two-word sequence of words like "could you"
Tri-gram is a three-word sequence of words like “could you explain” or "you explain what." Estimate the predict of words or seeking words (Brown et al., 1992) based on the previous words according to a sample text using n-gram model. We’d expect to

find the pair of words. Isolated words (Bahl, Jelinek, and Mercer, 1990) are being distinct purposes of syntactic analysis in the previous words wk−1.

We are using some NLP techniques like Tokenization, stemming, stopwords.

2.4 Term frequency - Inverse document frequency(TF- IDF)

Term frequency - Inverse document frequency (TF-IDF) is the weight of a term implies that how important is for a document in a collection or corpus. We have used TF-IDF to determine the term depends on relevance score of the whole or (in part) documents of each query term that appears in that document.
As TF- IDF has used several times for this thesis, the general idea of TF- IDF is described in this section. TF-IDF of a term t in a document d is calculated as follows:

$$TF - IDF_{t,d} = termFreq_{t,d} \times inDocFreq_t$$

Here termFreqt,d is the term frequency of a term t for the document d and inDocFreqt is the inverse document frequency of the term t.
Term frequency component implies a term appears in a number of times in the document. Inverse document frequency component signifies the overall weighted of a term in a given document.

How many times a term appears in a document is the t f value of a term in the document. The document d and term t of term frequency is followed by:

$$termFreq_{t,d} = \frac{Thetermtfrequencyappearsindocumentd}{Estimatethetermsinthecorpus}$$

Inverse document frequency(id f) defines how importance does the term appear in all documents. We have applied a python medium-scales named Scikit-learn (Pedregosa et al., 2011) to calculate TF-IDF where inDocFreqt is calculated the following equation.

$$inDocFreq_t = 1 + log(\frac{numDoc}{(1 + docFreq_{(d,t)})})$$

Where numDoc is the number of documents in the corpus and docFreq(d,t) is the document frequency which contains the term belongs to the documents.
If a term t appears the length of document d, higher weight or shorter length of the field, we normalized the field length as follows...

$$normDoc_{(nd)} = \frac{1}{\sqrt{lenTerms_t}}$$

While normDoc(nd) efficient for searching a query in the index.
Term frequency, Inverse document frequency, and normalized documents both are stored index time within a document regards of fundamental semantic (Blei, Ng, and Jordan, 2003) concepts.

2.5 Vector Space Model

Vector Space Model (VSM), text document representation such as provide automated information to the user (Salton, Wong, and Yang, 1975), store accurate and facilitate for designing the search engines, each number in the vector is the weight of a term as calculated among term frequency/inverse document frequency. Vector-based methods for performing query retrieval produces the high-quality search results (Ramos, 2003). We have used VSM to extract the query against the multi-term documents. It seems that vector representation both query and documentation.

VSM is the most significant and plotted algorithm that processes the user-defined query become such common and uncommon terms. We assume that common terms are very low weighted while different terms are relatively high weighted.

3. Design Approach

This approach to developing tutorial component is based on video targeted comments. Here, we pay more attention to extract comments on the selected online video learning. In our research, we analyzed to use each of the ap- proaches to comments collections. Figure 3.1 presents the overall prototype architecture of finding the answer to the question.

Our aim to explore the approach to extract the comments that apply to general users. We are dealing with comments similar to question and answer. We make use of recently archived dataset obtained from the learning website (Khan Academy) comments to extract the relevant question and answer. We have shown that web-based learning comments of data are useful at finding the information known by providing information that frequently occurs with the query terms in the context. Q/A systems that have been applied to data collections before, and it entirely relies on the question and answer of the tutoring videos.

Our design approaches focus on the analytical tools of the system architecture on heuristics. Some parts of our design work are more constrained by presentiment than others. Video-based learning platforms such as Khan Academy are of great value for informal learning on demand. Users can browse an extensive collection of videos and video-based learning courses in various domains to acquire specific knowledge in a self-directed manner. In addition to the videos and courses, Khan Academy supports knowledge exchange among the users. Each video has a question and answer thread where users can ask questions that are answered by the community.

From this, an extensive collection of questions and corresponding answers has emerged over time such that it is likely that for a given question in a specific area there exist several documents that have already been answered in the Q/A threads of Khan Academy videos. Since Q/A pairs are always linked to a single video, it is difficult to retrieve potential answers to questions across different videos.

Thus, the basic idea of this thesis is to use the vast corpus of question-answer pairs as a knowledge base to retrieve relevant answers given a user-defined question. To do this, a given free text question has to be matched to the existing questions in the database based on similarity measures. The solution should be integrated into a web platform users can use to find answers and related learning videos by providing a specific question.
This approach to developing a tutorial component is based on video targeted comments. Here, we pay more attention to extract comments on the selected online video learning. In our research, we analyzed to use each of the approaches to comments collections. Figure 3.1 presents the overall prototype architecture of finding the answer to the question.

4. Implementation

To implement extract and retrieve the documents from specific web-based learning data configured through elasticsearch REST APIs and also to evaluate the results of retrieving relevant documents we have used python as a programming language. Besides the built-in library of Python, we have used several third-party modules. They are: sklearn scikit- learn: ("Scikit-learn: Machine learning in Python" 2018), numpy(NumPy 2018), scipy (SciPy.org 2018), spacy (spaCy - Industrial-strength Natural Language Processing in Python 2018) and stopwords a widely used Python library named NLTK (Bird and Loper, 2004) library. The source code of our implementation and manually annotated datasets are available online.

The process starts with searching library namely Elasticsearch customize the searching techniques. In web-based learning like Khan Academy, people trend to comments on video lecture. We extract the comments using web crawling. Elasticsearch has several analyzers but we used the "standard analyzer " in English with a broad variety of built-in-analyzers which divides the text into terms. It removes the most punctuation, lowercases terms, supports to remove the stop words, and stemming.

Elasticsearch allows us to customize the analyzer process. We configured the built-in analyzers. We customized the stopwords removal list from the query with lowercased. Get tokens from a provided text. In our settings, the method provided uni-gram tokens. We also provide remove stopwords from the query. After that, we get a list of the query by removing one token for each time.

5. Evaluation and Results

For Evaluation, firstly, we took the 500 video comments from Khan Academy and pre-processed to only questions shown in figure 5.1. We made the 100 manual queries, described in section 5.2 by reading through the documents and two different raters annotated the relevant documents.

We have calculated the inter-rater reliability for making the data Ground truth data (GTD) that measures the Cohen's Kappa value. We represent the descriptive statistics histogram to show the relevant items frequency. Finally, we calculated the Precision, Recall, and F1-score based on the ground truth data (GTD) and averaged all of the query results.

We compared our particular approaches using the Khan Academy archived comments. In this collection, there are several video topic comments data, and it was pre-processed into 500 question pairs, but human-generated question annotations are available

for comparing to automated annotations. To develop the process, we are using archived video comments for the next evaluation to match concepts in the desired response.
The fundamental theory is to take all queries from a user and consider all questions in that session. Then calculate the precision, recall, and f1-score based on the inter-rater-reliability method and find the average score of all the queries.

5.1 Manual annotation

For manual annotation, we had assigned aspects to each query with relevant documents. We collected the human-generated query annotations with relevant documents. For the final evaluation, we have considered 100 annotated query set from two different raters.

In this thesis paper, the name of these five video comments are referred to as
Computer Programming(Clarifying with Comments), Computer programming intro to HTML/CSS: Making Web Pages, Physics One-Dimensional Motion, Velocity/ Speed, Reading Pictographs comments. In the manual annotation, we have made 100 manual queries, reading through the documents and two different raters annotated the relevant documents.

Figure 5.3 represents the statistics of a histogram of the number of queries against the number of relevant documents based on human judgment.
According to our observation of documents, we have seen, particular topics were discussed in several questions. We have split each manually annotated question, generated an ID and assigned topics to each question. A topic is a group of the question and answers pairs or phrases that indicate what the question is talking about.

For example, consider the following queries collected from the dataset of Khan Academy comments: "What does parabola mean?" This query represents the topic parabola. We created an order that will help an annotator to put the relevant documents in the dataset. Figure- 5.2 shows the manual relevant query annotation of five particular topics.

5.2 Measure Cohen's Kappa

Cohen's Kappa measurement is used to determine the measure of agreement between two raters which is the number of agreement scores/ total scores. For calculating Cohen's Kappa value, we considered the 100 relevant annotated query from two different raters and took the common annotation ID's. At first, we merged the human annotated documents and then, common all ID's and measured Cohen's Kappa value.

In our thesis evaluation, we have considered the value range to be

0.5 - 1.00 = agreement equivalent to chance alone.
0.1 – 0.20 = slight agreement.
0.21 – 0.40 = fair agreement.
0.41 – 0.60 = moderate agreement.
0.61 – 0.80 = substantial agreement.
0.81 – 0.99 = near perfect agreement.
1.00 = perfect agreement.

We are representing the statistical histogram of Cohen's kappa value against the number of manual queries in figure 5.4 and the number of relevant documents based on the human judgment. In our inter-rater reliability (IRR) measures, we considered >= 0.5 Cohen's Kappa agreement of each query values. We calculated the Precision, Recall, and F1-score of each query if Cohen's kappa agreement value is >= 0.5.

We took the common two raters annotated relevant queries and measured the Cohen's Kappa value for ground truth (major agreement) data. A document is considered to be relevant when all raters judged it as relevant. Then calculated the precision, recall and, f1-score for agreed values.
We evaluated the different background set by firstly performing the agreement data and considered the manually annotated dataset. Secondly, we perform the precision, recall, and f1-score of each query based on the agreement data. In the third configuration, we performed the average performance of the systems by combining the single searching query. We also performed the stopwords removal and domain-specific words.

This section contains the result of evaluation and it manifests the effectiveness of our method. Standard stopwords list and after modification, we get the different accuracy of the system. For any decision to be made by the system performance, we'll show the effect of some results.
For evaluation, we considered 100 manual queries, but we show a sample of 10 query results. We don't show all of the query results in this write-up.
We have single search results versus combined search results. Standard stopwords list result versus domain specific list words result. We get the different effect of evaluation results.

Results of our agreement data measured by Cohen's Kappa in table 5.1 shows the values we get from the manually annotated relevant query described in section 5.2 by two different raters. As a result of this, we represent only 10 sample queries result.

To handle the ground truth (GT) data for agreement, we manually reported the number of relevant data for a sample of 100 queries. For precision, recall, and f1-score values, we considered the Cohen's Kappa values whose agreement values are more than 0.5. After the agreement, we got 84 major agreement queries which values are more than 0.5. Table 5.2 and 5.3 shows the effects of precision and recall values respectively.

We merely take the precision and recall of each query based on our agreement data. If we miss some relevant document, in that situation precision might be 100 percent, but recall might not be good.

The raters contributed a total of 100 queries. Each query was informational; it seems to evaluate the performance of the systems. The Cohen's Kappa measures the percentage of foresight that equals to any one of the ground truth data. The f1-score measures the average overlap between the prediction and ground truth agreement. Then it is closer to the smaller number as compared to the more significant amount shown in table 5.4 for each query.

We evaluated the reliability of our query set by comparing it to the manually annotated documents based on the inter-rater reliability. We assessed the quality of our system performance based on the mean precision, recall and f1-score.

The average value for a single query shown in table 5.6 is calculated by taking the mean of the precision, recall, and f1-score. The mean precision value for a single query is calculated by taking the mean of the precision score after retrieving the manual document. Table 5.5 contains the overall average performance of the quality of combining query search results.

The final thesis work must be the focus on the value, the search value can be the proper evaluation. We evaluated the distinctive values of queries. We use the standard stopwords removal list to evaluate the results. Either it could improve the result or we have to make a certain decision about how can we improve the system performance. Our focus is on domain-specific concepts. For precision value, we get the sound result to use domain-specific words than the standard stopwords list like what, why, relate, where. But recall value isn't better.

We compare the result of standards stopwords list and with domain-specific words shown in table 5.7 and 5.8. If we miss some relevant documents in that situation precision might be 100 percent, but recall might not be accurate. We didn't consider the rewrite methodology because we applied English analyzer on "question" index for pre-configuration for Elasticsearch. Analyzer provides the n-gram model.

The overall result of all the manual datasets is shown in table 5.9. From the overall consequence shown in table 5.9, it could be noted that the system accuracy from the single searching result performs worst while combined searching result performs the best.

6. Summary

In our research, we aimed to determine to find useful textual information and analyze the domain and signal concepts while learning one or more topics. In this thesis paper, we develop a tutorial component to extract the video targeted comments from the online discussion. To approach the extraction of the implicit question and answer comments. We use the Khan Academy online discussion comments to compute the similarities degree.
In this thesis paper, we have described extracting comments on online video questions and answers data and then generating the review of the label for each question and answer pair.

We have used the default elasticsearch string-distance algorithm measure entirely as the distance function. To make a sentence vector for a sentence from its words, we have estimated removing stop words from the sentence. In the evaluation section, we have found, perform the average value of the relevant data. We have considered a set of re-phrases as labels of combination words. We have been initialized by determining the set of essential uni-gram terms from the documents. To find important terms that we have considered the following methods.

In our first method, the text of the documents has been tokenized into unigram terms. Then stop-words from terms have been removed, and we have implemented stemming to every remaining term. After applying to stem to terms, similarly, we have been represented the syntactically related words. Then the TF-IDF weighting scheme has been weighted in each term. where TF has been calculated based on given the term appear in the document and IDF has been determined from the term appear in all documents in the collection. IDF value of a term has remained the same in common terms which contribute to the relevance of words, as they appear in most documents.

In the second approach, documents of the texts we also been tokenized into uni-gram terms. Similarly, to handle related terms using the stemming While calculating the IDF value of a term, we have taken similar words. We have calculated similar words, each word has the same IDF value. To calculate the TF value of the term, we have replaced all words in a term-document with the one that has appeared the higher the weight in the document.

In the third, we have used to domain concepts to extract keywords from a document. Top-weighted terms have been considered as important terms. Besides uni-gram terms those were extracted using the several approaches, we have used noun phrases have been extracted from the document. If every word in a noun phrase appeared in top terms, the noun phrase has been estimated as a higher weight of the documents and the noun phrase including words have been removed from candidate label a set of question. Stop words have not been considered before splitting a noun phrase. In first and third methods, we applied to stem to each word of a noun phrase.

For the evaluation, we have used fraction relevant results evaluation for using archived comments. Precision-recall is the criteria for relevant evaluation where the relevant and irrelevant of each sentence from the document. To evaluate the result, how well the search is satisfied, we have matched set of the generated question with the set of all documents. The effect of each method measures the retrieval results. F1-score is the measure of a test's accuracy in weighted of precision-recall. In every case, all of our evaluation methods have exceeded the average the retrieval of relevant answers.

According to the result of table 5.9, domain-specific words return the best result whereas stopwords return the less performance instead of measuring words which are the combination of the different patterns. Table 5.9 shows the overall system performance with several techniques.

6.1 Future work

One of the limitations of our approach in both documents and query is using pre-trained NLP using NLTK (Bird and Loper, 2004) model. We have used the pre-trained model. The model provides outstanding results due to its optimization. However, every word in our corpus is not happened to appear in the pre-trained NLTK model. With the method, many essential words might not be in considerations. In our approach there are synonyms words doesn't work. To address this issue, in the future work, We can train and optimize the NLTK model with a large number of video comments. The several sources of data might also be filled to train the NLTK model to get a better outcome.
To our knowledge, there is a more standard or well-established method to evaluate information extraction. We have shown a way to evaluate our query with the pre-annotated relevant query. As our evaluation process is not a standard one, the technique could have some side effects. Manual evaluation of query could be a recovering.

List of Figures

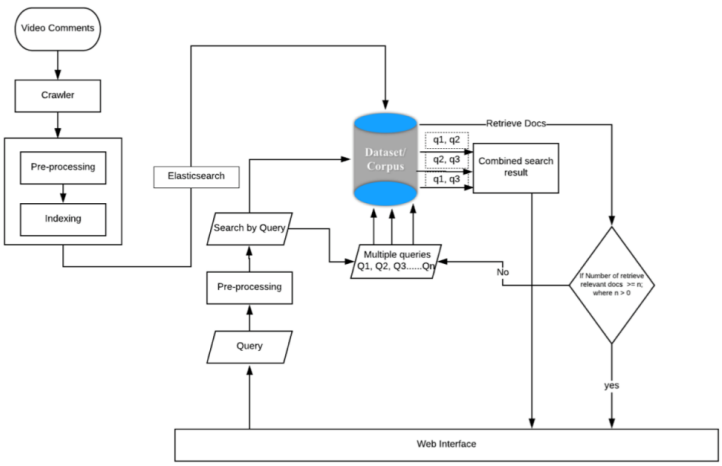

FIGURE 3.1: Overall prototype architecture of finding the answer to question

```
{
  "0": "What about the blue comments (The doc comments). We can use them, too, right?  Or are they not
  really supposed to be used in this way? (The doc comments start with `/**` and end with `*/`.)",
  "1": "So comments are basically like Pseudo-Code in the previous video?\nThis kinda confuses me.",
  "2": "I have done javaScript in codecademy.com and in Khan Academy. And I realized that these two
  javaScripts are different(!?). Could anyone tell me why are they different?",
  "3": "How do you do a star?",
  "4": "Can you do other colors on comments than just blue and green? \nhow about orange",
  "5": "Why when I put the words, \"`TODO`\" into a blue comment (`/** TODO **/`), the words (TODO) turn
  into a green color? Does it signify anything? Is it a glitch? Another Khan Academy function I don't know
  about?\nThanks!",
  "6": "So the comments are just a way of explaining things to other programmers and don't modify the code
  itself?",
  "7": "What does parbola mean?",
  "8": "can you trick out comments to be any color?",
  "9": "how in the world did all those comments appear after Pamela highlighted it?",
  "10": "You talked all about green comments. What are blue comments for?",
  "11": "what is the difference between:\n/*\n*/\nand\n/**\n**/\n?",
  "12": "is better to put a comment like this\n(coding) //(comment)//\nor\n //(comment)//\n(coding)",
  "13": "what is the difference between using 2 stars like: ```/**hi**/  ```and 1 star like: ```/* hi*/
  ```?",
  "14": "I've browsed other programs on Khan Academy and seen some really advanced ones. I did notice that
  some people have little bars, or buttons, within their code that essentially hide their code to save
  space. What is that trick called and how do they do that?",
  "15": "// Calculate y according to the path of a parabola with respect to \"t\":\n   // y = (velocity)*t
  - (acceleration)*t^2;\n   // We can tweak the velocity and acceleration constants for our fake world\n
  y = -0.02*t+t+ 3.5*t;\nIn this formula of y if I use -0.02t instead of t the ball never comes back and
  if I use t^2 the ball bounces faster than t*t but t+t = t^2. Also, if I use only t^ the ball still
  bounces. can anyone explain why?"
}
```


FIGURE 5.1: A sample of only question set to find the relevant documents


```
{
"what is the definition of magnitude":[142, 103, 127, 133, 139, 146, 152, 154, 161, 184, 159, 160],
"Difference between speed and velocity": [102, 189, 130, 168, 159, 191, 105, 166, 131],
"What is the difference between /** and /*":[220, 109, 199, 2, 21, 49, 52, 83, 86],
"What about the blue comments": [28, 19, 69, 10, 0, 4, 94, 35, 5, 6, 9, 10, 18, 91, 94, 8, 26, 92, 20, 33,
31, 19, 69, 35, 26, 10],
"How could use a comment": [61, 0, 18, 26, 31, 46, 28, 69, 10, 44, 45, 46],
" How to make bouncing ball using computer programming": [79, 91, 16, 57, 23, 74, 96, 97, 98],
"How to animate rain drops using JS": [80],
"What is parabola mean": [60, 42, 32, 148, 15, 7, 32, 42],
"Difference between mass and weight": [372, 381, 333, 367, 383, 354, 318, 372, 387, 5],
"What is Atomic Weight and Atomic Mass": [382, 381, 378, 384, 305, 367, 328, 300, 306, 354, 318, 382, 383],
"how many applications using by pictograph": [474 ],
"How to reveal the comments for highlighting the code":[61],
"The main difference between scalars and vectors":[132],
" What is a vector and a scalar quantity":[193, 180, 111, 147, 125, 104, 186, 127, 156, 129, 102, 118, 192,
145,136, 164, 106, 134, 180, 170, 122, 132],
"Is a time vector or  scalar or represent both":[ 156, 125, 107,111, 128, 180, 186],
"Does the work is a scalar in  physics and consider the work as a scalar or a vector":[136, 118],
"How to use HTML in javascript": [43, 294, 214, 230, 2, 227, 274, 248, 253, 250],
"What is an isotope": [326, 371, 377, 313, 357, 350, 342, 381, 316, 386, 336, 317, 331],
"What are the - and + on the blood types": [433, 439, 429, 412, 496 ],
"how many blood groups are available": [427, 460, 467, 439, 433, 450, 436, 234, 291, 213, 257, 295, 296,
202],
"What's the relationship between a server and a client": [204, 264, 282]
}
```


FIGURE 5.2: A sample of manual relevant question annotation

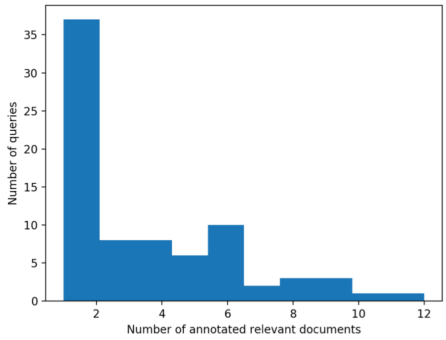

FIGURE 5.3: Statistics in a histogram of the number of queries against the number of annotated relevant documents

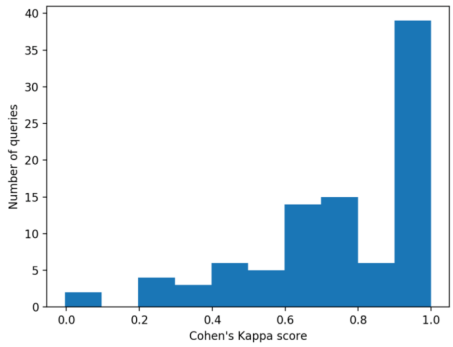

FIGURE 5.4: Statistics histogram of number of queries and Cohen's Kappa value
```

List of tables

Query	Cohen's Kappa Value
Difference between speed and velocity	0.711
What is parabola mean	0.798
Where comes from a pictograph	0.549
Is Displacement different from Distance	0.821
Does mean the Atomic weight is an average weight of all the atoms in the world	0.625
What is the difference between velocity and displacement	0.799
How can carbon an isotope stable state	1.0
How to know electron in atoms	0.664
Why human feel to acceleration and deceleration	0.666
Does the work is a scalar in physics and consider the work as a scalar or a vector	1.0

TABLE 5.1: Results of our agreement data measured by Cohen's Kappa

Query	Recall
Difference between speed and velocity	0.2
What is parabola mean	0.75
Where comes from a pictograph	0.4
Is Displacement different from Distance	0.858
Does mean the Atomic weight is an average weight of all the atoms in the world	0.5
What is the difference between velocity and displacement	0.5
How can carbon an isotope stable state	0.5
How to know electron in atoms	0.667
Why human feel to acceleration and deceleration	1.0
Does the work is a scalar in physics and consider the work as a scalar or a vector	0.5

TABLE 5.3: The recall value of each query

Query	Precision
Difference between speed and velocity	0.333
What is parabola mean	1.0
Where comes from a pictograph	0.2
Is Displacement different from Distance	0.75
Does mean the Atomic weight is an average weight of all the atoms in the world	0.3333
What is the difference between velocity and displacement	1.0
How can carbon an isotope stable state	1.0
How to know electron in atoms	0.5
Why human feel to acceleration and deceleration	1.0
Does the work is a scalar in physics and consider the work as a scalar or a vector	1.0

TABLE 5.2: The precision value of each query

Query	F1-score
Difference between speed and velocity	0.25
What is parabola mean	0.667
Where comes from a pictograph	0.267
Is Displacement different from Distance	0.799
Does mean the Atomic weight is an average weight of all the atoms in the world	0.4
What is the difference between velocity and displacement	0.5
How can carbon an isotope stable state	0.667
How to know electron in atoms	0.571
Why human feel to acceleration and deceleration	1.0
Does the work is a scalar in physics and consider the work as a scalar or a vector	0.667

TABLE 5.4: The f1-score of each query

Mean Precision	0.776
Mean Recall	0.681
Mean F1-score	0.685

TABLE 5.5: The combined search results

Mean Precision	0.604
Mean Recall	0.514
Mean F1-score	0.521

TABLE 5.6: Search results with single query

Mean Precision	0.721
Mean Recall	0.709
Mean F1-score	0.682

TABLE 5.7: Result of standard stopwords remove list

Mean Precision	0.774
Mean Recall	0.674
Mean F1-score	0.679

TABLE 5.8: Result of domain-specific words

Measure	Combine	Single	Stopwords removal	Domain-specific
Mean precision	0.776	0.604	0.721	0.774
Mean recall	0.681	0.514	0.709	0.674
Mean f1-score	0.685	0.521	0.682	0.679

TABLE 5.9: Overall result of precision, recall, and f1-score using different methods

Bibliography

Aker, Ahmet et al. (2016). "Automatic label generation for news comment clusters". In: *Proceedings of the 9th International Natural Language Generation conference*, pp. 61–69.

Bahl, Lalit R, Frederick Jelinek, and Robert L Mercer (1990). "A maximum likelihood approach to continuous speech recognition". In: *Readings in speech recognition*. Elsevier, pp. 308–319.

Bai, Jun (2013). "Feasibility analysis of big log data real time search based on hbase and elasticsearch". In: *Natural Computation (ICNC), 2013 Ninth International Conference on*. IEEE, pp. 1166–1170.

Bird, Steven and Edward Loper (2004). "NLTK: the natural language toolkit". In: *Proceedings of the ACL 2004 on Interactive poster and demonstration ses- sions*. Association for Computational Linguistics, p. 31.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022. Bourigault, Didier (1992). "Surface grammatical analysis for the extraction of terminological noun phrases". In: *Proceedings of the 14th conference on Com- putational linguistics-Volume 3*. Association for Computational Linguistics, pp. 977–981.

Boyer, Robert S and J Strother Moore (1977). "A fast string searching algo- rithm". In: *Communications of the ACM* 20.10, pp. 762–772.

Brooks, Christopher A et al. (2011). "OpenCast Matterhorn 1.1: reaching new heights". In: *Proceedings of the 19th ACM international conference on Multi- media*. ACM, pp. 703–706.

Brown, Peter F et al. (1992). "Class-based n-gram models of natural language". In: *Computational linguistics* 18.4, pp. 467–479.

Charras, Christian, Thierry Lecrog, and Joseph Daniel Pehoushek (1998). "A very fast string matching algorithm for small alphabets and long pat- terns". In: *Annual Symposium on Combinatorial Pattern Matching*. Springer, pp. 55–64.

Chen, Stanley F and Joshua Goodman (1999). "An empirical study of smooth- ing techniques for language modeling". In: *Computer Speech & Language* 13.4, pp. 359–394.

Clarke, Clarlie LA and Gordon V Cormack (1995). "Dynamic inverted in- dexes for a distributed full-text retrieval system". In: *Ws ec u R eport CS- 95-0* 10.

Cutting, Doug and Jan Pedersen (1989). "Optimization for dynamic inverted index maintenance". In: *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 405–411.

Daems, Oliver et al. (2014). "Using content analysis and domain ontologies to check learners' understanding of science concepts". In: *journal of com- puters in education* 1.2-3, pp. 113–131.

Deshpande, Anjali R and LMRJ Lobo (2013). "Text summarization using clus- tering technique". In: *International Journal of Engineering Trends and Tech- nology* 4.8.

Divya, Manda Sai and Shiv Kumar Goyal (2013). "ElasticSearch: An advanced and quick search technique to handle voluminous data". In: *Compusoft* 2.6, p. 171.

Elasticsearch. "https://www.elastic.co/". In:

Fan, Weiguo et al. (2006). "Tapping the power of text mining". In: *Communications of the ACM* 49.9, pp. 76–82.

Freedman, Reva, Syed S Ali, and Susan McRoy (2000). "Links: what is an intelligent tutoring system?" In: *intelligence* 11.3, pp. 15–16.

Gaikwad, Sonali Vijay, Archana Chaugule, and Pramod Patil (2014). "Text mining methods and techniques". In: *International Journal of Computer Applications* 85.17.

Gao, Li, Elizabeth Chang, and Song Han (2005). "Powerful tool to expand business intelligence: Text mining". In: *Transactions on ENFORMATIKA, Systems Sciences and Engineering, vol. 8, ESSE 2005, Budapest*. International Academy of Sciences, pp. 110–115.

Giannakos, Michail N (2013). "Exploring the video-based learning research: A review of the literature". In: *British Journal of Educational Technology* 44.6. Giannakos, Michail N and Panayiotis Vlamos (2013). "Using webcasts in ed- ucation: Evaluation of its effectiveness". In: *British Journal of Educational Technology* 44.3, pp. 432–441.

Google. "http://www.google.com". In:

Gormley, Clinton and Zachary Tong (2015). *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. " O'Reilly Media, Inc." Gottipati, Swapna, David Lo, and Jing Jiang (2011). "Finding relevant an- swers in software forums". In: *Proceedings of the 2011 26th IEEE/ACM In- ternational Conference on Automated Software Engineering*. IEEE Computer Society, pp. 323–332.

Greenwood, Mark Andrew (2005). "Open-domain question answering". PhD thesis. University of Sheffield, UK.

Gupta, Vishal, Gurpreet S Lehal, et al. (2009). "A survey of text mining techniques and applications". In: *Journal of emerging technologies in web intelligence* 1.1, pp. 60–76.

Hallgren, Kevin A (2012). "Computing inter-rater reliability for observational data: an overview and tutorial". In: *Tutorials in quantitative methods for psy-*

chology 8.1, p. 23.
Harris, Howard and Sungmin Park (2008). "Educational usages of podcast-
ing". In: *British Journal of Educational Technology* 39.3, pp. 548–551. Higashinaka, Ryuichiro et al. (2013). "Question answering technology for pinpointing answers to a wide range of questions". In: *NTT Technical Re-
view* 11.7.
Hovy, Eduard H, Ulf Hermjakob, Chin-Yew Lin, et al. (2001). "The Use of External Knowledge of Factoid QA." In: *TREC*. Vol. 2001, pp. 644–52.
Iiritano, S and Massimo Ruffolo (2001). "Managing the knowledge contained in electronic documents: a clustering method for text mining". In: *Database and Expert Systems Applications, 2001. Proceedings. 12th International Work- shop on*. IEEE, pp. 454–458.
Isozaki, Hideki et al. (2009). *Question answering systems*.
Janani, R and S Vijayarani (2016). "An Efficient Text Pattern Matching Al-
gorithm for Retrieving Information from Desktop". In: *Indian Journal of
Science and Technology* 9.43.
Jurafsky, D and JH Martin (2014). "Dialog Systems and Chatbots". In: *Speech
and language processing* 3.
Jurafsky, Dan and James H Martin (2009). *Speech and language processing: An
introduction to natural language processing, computational linguistics, and speech
recognition*.
Kaplan, Ronald M (1972). "Augmented transition networks as psychological
models of sentence comprehension". In: *Artificial Intelligence* 3, pp. 77–100. Kiewitt, Eva L (1979). *Evaluating information re-
trieval systems: the PROBE pro-
gram*. Greenwood Publishing Group Inc.
Kiss, Tibor and Jan Strunk (2006). "Unsupervised multilingual sentence bound-
ary detection". In: *Computational Linguistics* 32.4, pp. 485–525. Kononenko, Oleksii et al. (2014). "Mining modern repositories with elastic- search". In: *Proceedings of the 11th Working Conference on Mining Software
Repositories*. ACM, pp. 328–331.
Krallinger, Martin and Alfonso Valencia (2005). "Text-mining and information-
retrieval services for molecular biology". In: *Genome biology* 6.7, p. 224. Kurz, Daniela and Feiyu Xu (2002). "Text mining for the extraction of domain relevant terms and term collocations". In: *Proceedings of the International
Workshop on Computational Approaches to Collocations*. Citeseer.
Larson, Ray R (2010). *Introduction to information retrieval*.
Manning, Christopher D, Prabhakar Raghavan, and Hinrich Schutze. "Intro-
duction to Information Retrieval‖ Cambridge University Press, 2008". In:
Ch 20, pp. 405–416.
Martin, Fred G (2012). "Will massive open online courses change how we
teach?" In: *Communications of the ACM* 55.8, pp. 26–28.
Miner, Gary, John Elder IV, and Thomas Hill (2012). *Practical text mining and
statistical analysis for non-structured text data applications*. Academic Press. Morovic´, Srec´ko and Salvatore Zaza (2015). "A scalable monitoring for the
CMS Filter Farm based on elasticsearch". In:
Navathe, Shamkant B and Elmasri Ramez (2000). "Data warehousing and
data mining". In: *Fundamentals of Database Systems*, pp. 841–872. Padhiyar, Hiral D and Dilipsinh N Padhiar. "Improving Accuracy of Text
Classification for SMS Data". In:
Patel, Mr Rahul and Mr Gaurav Sharma (2014). "A survey on text mining
techniques". In: *International Journal Of Engineering And Computer Science ISSN* 2319.7242, pp. 5621–5625.
Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.
Pitts, Anna (2013). *Learning is multi-sensory: How to engage all the senses so chil- dren really benefit*.
Porter, Martin F (1980). "An algorithm for suffix stripping". In: *Program* 14.3, pp. 130–137.
Rahman, Aedah Abd, Munaisyah Abdullah, and Siti Hajar Alias (2016). "The architecture of agent-based intelligent tutoring system for the learning of software engineering function point metrics". In: *Agent, Multi-Agent Systems and Robotics (ISAMSR), 2016 2nd International Symposium on*. IEEE, pp. 139–144.
Rajman, Martin and Romaric Besançon (1998). "Text mining: natural lan- guage techniques and text mining applications". In: *Data mining and re- verse engineering*. Springer, pp. 50–64.
Ramos, Juan et al. (2003). "Using tf-idf to determine word relevance in docu- ment queries". In: *Proceedings of the first instructional conference on machine learning*. Vol. 242, pp. 133–142.
Reelsen, Alexander (2014). "Using elasticsearch, logstash and kibana to cre- ate realtime dashboards". In: *Dostupné z: https:// secure. trifork. com/dl/goto- berlin-2014/GOTO_Night/logstash-kibanaintro. pdf*.
Riley, Michael Dennis (2013). *Local item extraction*. US Patent 8,433,704. Sadamitsu, Kugatsu et al. (2013). "Knowledge extraction from text for intel-

ligent responses". In: *NTT Technical Review* 11.7, pp. 1–5.
Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620.
Sampson, Demetrios G et al. (2014). "Digital systems for open access to formal and informal learning". In: *Digital systems for open access to formal and informal learning*. Springer, pp. 1–7.
"Scikit-learn: Machine learning in Python" (2018). In:
Siemens, George (2003). "Learning ecology, communities, and networks: Extending the Classroom. Elearnspace". In: *Everything eLearning*.
— (2005). "Connectivism: A learning theory for the digital age". In: *International journal of instructional technology and distance learning* 2.1, pp. 3–10. Song, Hui et al. (2013). "Semantic Analysis and Implicit Target Extraction of Comments from E-Commerce Websites". In: *Software Engineering (WCSE), 2013 Fourth World Congress on*. IEEE, pp. 331–335.
Sukanya, M and S Biruntha (2012). "Techniques on text mining". In: *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on*. IEEE, pp. 269–271.
Syeda-Mahmood, Tanveer et al. (2005). "Searching service repositories by combining semantic and ontological matching". In: *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, pp. 13– 20.
Tellex, Stefanie et al. (2003). "Quantitative evaluation of passage retrieval al- gorithms for question answering". In: *Proceedings of the 26th annual inter- national ACM SIGIR conference on Research and development in informaion retrieval*. ACM, pp. 41–47.
Tong, Hanghang and U Kang (2013). *Big Data Clustering.*
Voorhees, Ellen M and Dawn M Tice (2000). "Building a question answering test collection". In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 200–207.
Wenger, Etienne (1987). "Artificial intelligence and tutoring system: Com- putational and Cognitive Approaches to the Communication of Knowl- edge". In: *Califórnia: Morgan Kaufmann Publishers. Texto publicado na: Pátio- revista pedagógica Editora Artes Médicas Sul Ano* 1, pp. 19–21.
Woods, William A and R Kaplan (1977). "Lunar rocks in natural English: Ex- plorations in natural language question answering". In: *Linguistic struc- tures processing* 5, pp. 521–569.
Yang, Hui et al. (2003). "VideoQA: question answering on news video". In: *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, pp. 632–641.
Zhang, Zhiyong and Olfa Nasraoui (2006). "Mining search engine query logs for query recommendation". In: *Proceedings of the 15th international confer- ence on World Wide Web*. ACM, pp. 1039–1040.
Zucchetto, Jasonc´ (2004). "https://www.elastic.co/". In: