

A Synopsis

on

CLASSIFICATION OF YOU-TUBE VIDEOS USING DEEP LEARNING

in partial fulfillment of the requirement for the degree

of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Sivasish Koch(1513310214)

Saif Ali(1513321165)

Rahul Bilra(1513310161)

Akash Kandpal(1513310027)

under the supervision of

Dr. Deepti Gupta

(Associate Professor Computer Science Dept.)



**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY
GREATER NOIDA**

Index

Sr.No.	Topics	Page No.
1	Abstract	1
2	Introduction	2
3	Existing System(Literature Survey)	4
4	Problem Statement	7
5	Proposed Methodolgy	9
6	Objective of the project and Features	11
7	Resource Requirements	13
8	Conclusion	14
9	References	15

Supervisor Sign:
Sign:

Group Members

ABSTRACT

You-Tube contains lots of videos which are both good and bad in terms of the quality of the content and the satisfaction of the user. We search on YouTube for a video related to our projects and/or studies but end up getting distracted by advertisements, promotional videos, music videos or any other type of video and then after an hour or two we realize that we wasted our time and regret afterward. This **reduces** our **productivity** and also makes us **mentally tired** to watch more videos regarding our subject.

We are unable to get quality content from the pool of videos. So, we came across this idea of providing quality scores to videos (currently for education domain) and further provide tags to it for categorizing the videos in different domains.

INTRODUCTION

YouTube is the most popular and most used video platform in the world today. YouTube has [a list of trending videos](#) that is updated constantly. Here we will use Python with some packages like Pandas and Matplotlib to analyze a dataset that was collected over 205 days. For each of those days, the dataset contains data about the trending videos of that day. It contains data about more than 40,000 trending videos. We will analyze this data to get insights into YouTube trending videos, to see what is common between these videos. Those insights might also be used by people who want to increase popularity of their videos on YouTube.

The dataset that we will use is obtained from Kaggle [here](#). It contains data about trending videos for many countries. Here we will analyze USA trending videos.

Goals of the analysis

We want to answer questions like:

- How many views do our trending videos have? Do most of them have a large number of views? Is having a large number of views required for a video to become trending?
- The same questions above, but applied to likes and comment count instead of views.
- Which video remained the most on the trending-videos list?
- How many trending videos contain a fully-capitalized word in their titles?
- What are the lengths of trending video titles? Is this length related to the video becoming trendy?
- How are views, likes, dislikes, comment count, title length, and other attributes correlate with (relate to) each other? How are they connected?
- What are the most common words in trending video titles?

- Which YouTube channels have the largest number of trending videos?
- Which video category (e.g. Entertainment, Gaming, Comedy, etc.) has the largest number of trending videos?
- When were trending videos published? On which days of the week? at which times of the day?

MODIFICATION AND IMPROVEMENT OVER THE EXISTING IMPLEMENTATION

Present State:

Software present currently are limited in scope and they don't deal directly with improving the content of the You-Tube.

Currently the systems mainly focusses on building recommendation systems for You-Tube and like-wise video sites.

Till now You-Tube itself is not considering about categorizing the videos and checking the quality so an effort is required in this area.

After implementation of project:

Quality score for all videos can be seen in the You-Tube itself.

No need for an additional app and the user-experience with the You-Tube videos will be maintained.

User could save his/her time as he knows which videos are of good quality.

Building Recommendation videos will be easier after categorization of videos.

PROBLEM STATEMENT

You-Tube contains lots of videos which are both good and bad in terms of the quality of the content and the satisfaction of the user. We search on YouTube for a video related to something but end up getting distracted by advertisements, promotional videos, music videos or any other type of video and then after an hour or two we realize that we wasted our time and regret afterward. This **reduces** our **productivity** and also makes us **mentally tired** to watch more videos regarding our subject.

We are unable to get quality content from the pool of videos. So, we came across this idea of providing quality scores to videos and also provide tags to it for categorizing the videos in different domains. It can act as a self-checking mechanism and prevent us from getting distracted by unwanted and promotional videos. This will **increase productivity** and keep us on track and prevent mental tiredness. It can be used in **office environment** and in **institutions like colleges** .

OBJECTIVE

The following are the objectives of the A.I. :

1. Extracting comments on videos from You-Tube and using them to generate score for the video. The higher the score the better the video.
2. ML model will be able to read and understand the comments and it generates a list of words used in the positive comments and negative comments with respect to the title of the video for generating quality scores.
3. Simple approach of number of cumulative matches with the title is considered for the score. Also, we will produce list of words which belong to a particular domain.
4. Saving the time of users by providing them with the quality score for each video.
5. Till now, we are more dependent on the number of views for the quality of the video but we have considered a few factors like comments(textual analysis), views and video itself(in future) for providing score to a video.
6. Improving the quality of You-Tube as a video platform.

This project is mainly used by two types of users :

- i. Daily You-Tube users
- ii. You-Tube developers for recommendation systems on the user data.

Here are the respective modules.

S.No	Module Name	Percentage
1	Scraper for comments from Educative Videos(Backend)	100
2	Clean the comments (removing stop-words) and manual cleaning (Backend)	100
3	Get titles and description of videos with more educative content and save their data in list of words (Backend)	100
4	ML Model for classifying these comments, using positive and negative txt files(Backend)	100
5	Script for Fetch the titles and description and match with the list of words txt file. (Backend)	100
6	Generate the score for education videos using separate script. (Backend)	100
7	AJAX script for rendering server calls(Front-End)	100
8	Deep NLP research paper implemenation	0

- Cleaning and analysis of comments is done on manually selected You-Tube videos on education. We are collecting the most found words from the video and using them for generating the score about the educative content present in the video.
- JS scripts and AJAX are written for showing the generated score back on the screen.
- Sentiment analysis modules has been used for analysing the comments and telling whether it should be educative video or not.
- Scraping the comments using Selenium chromedriver.
- FB's fasttext library has been used for word-corpora and positive-negative

words list has been created using manually.

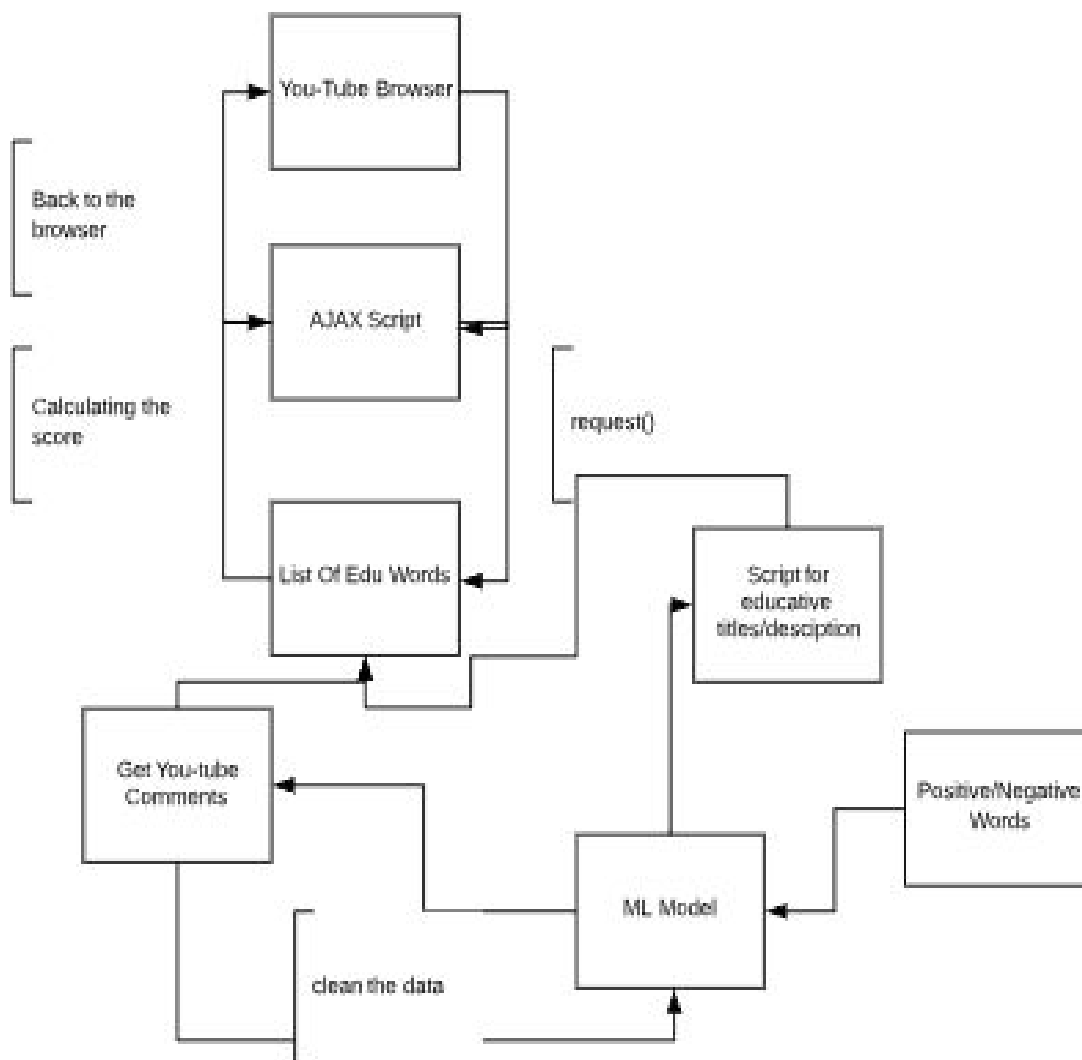
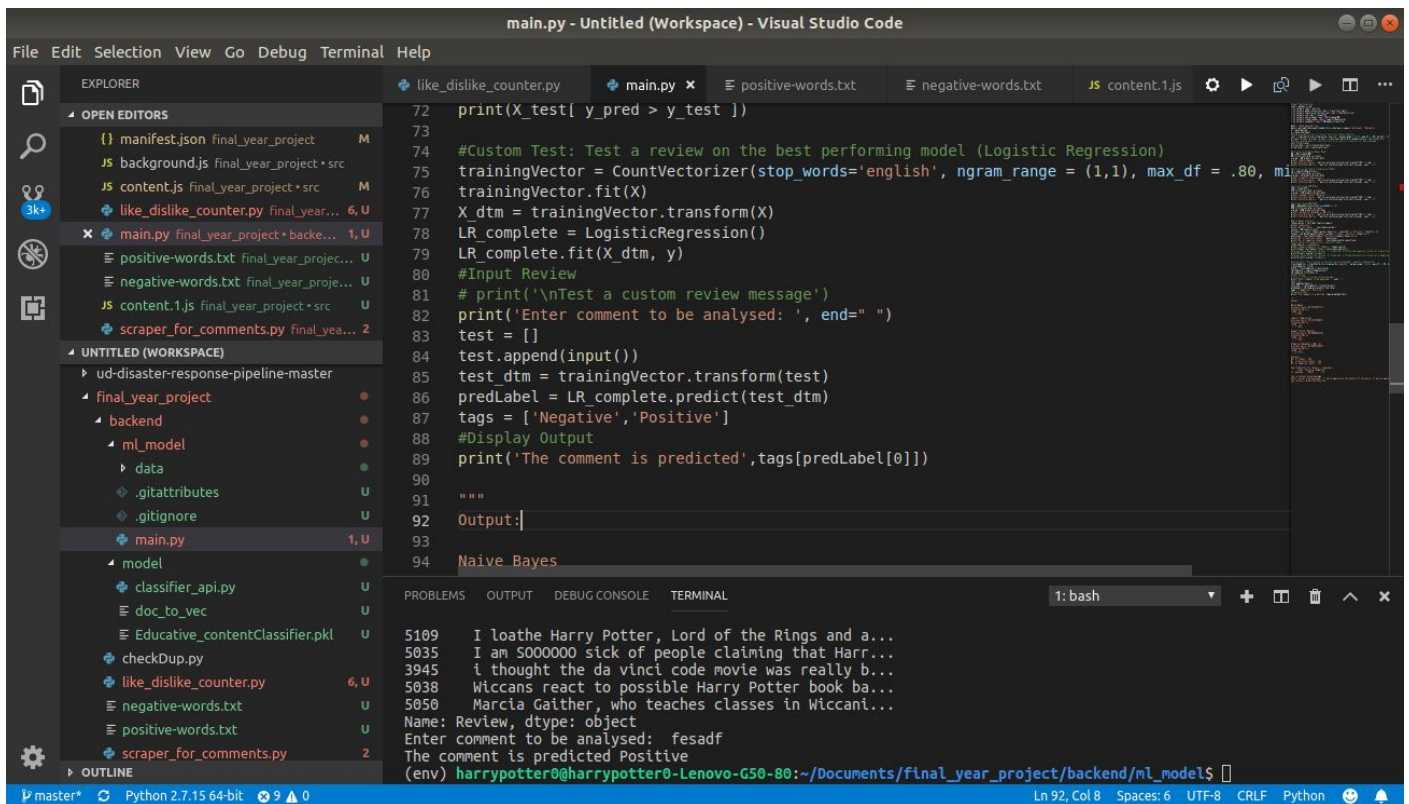


Figure : DFD Diagram

- Scraper for comments from Educative Videos(Backend).
- Clean the comments (removing stop-words) and manual cleaning (Backend).
- Get titles and description of videos with more educative content and save their data in list of words (Backend).
- ML Model for classifying these comments, using positive and negative txt

files(Backend).

- Script for Fetch the titles and description and match with the list of words txt file. (Backend).
- Generate the score for education videos using separate script. (Backend).
- AJAX script for rendering server calls(Front-End).
- Deep NLP research paper implemenation.



```
main.py - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

EXPLORER
  OPEN EDITORS
    {} manifest.json final_year_project M
    JS background.js final_year_project+src
    JS content.1.js final_year_project+src M
    like_dislike_counter.py final_year... 6,U
    x main.py final_year_project+backe... 1,U
    positive-words.txt final_year_projec... U
    negative-words.txt final_year_projec... U
    JS content.1.js final_year_project+src
    scraper_for_comments.py final_yea... 2
  UNTITLED (WORKSPACE)
    ud-disaster-response-pipeline-master
    final_year_project
    backend
      ml_model
        data
        .gitattributes
        .gitignore
        main.py 1,U
        model
          classifier_api.py U
          doc_to_vec U
          Educative_contentClassifier.pkl U
          checkDup.py
          like_dislike_counter.py 6,U
          negative-words.txt U
          positive-words.txt U
          scraper_for_comments.py 2
      OUTLINE

main.py
72 print(X_test[ y_pred > y_test ])
73
74 #Custom Test: Test a review on the best performing model (Logistic Regression)
75 trainingVector = CountVectorizer(stop_words='english', ngram_range = (1,1), max_df = .80, min_df = .20)
76 trainingVector.fit(X)
77 X_dtm = trainingVector.transform(X)
78 LR_complete = LogisticRegression()
79 LR_complete.fit(X_dtm, y)
80 #Input Review
81 # print('\nTest a custom review message')
82 print('Enter comment to be analysed: ', end=" ")
83 test = []
84 test.append(input())
85 test_dtm = trainingVector.transform(test)
86 predLabel = LR_complete.predict(test_dtm)
87 tags = ['Negative','Positive']
88 #Display Output
89 print('The comment is predicted',tags[predLabel[0]])
90
91 """
92 Output:
93
94 Naive Bayes

TERMINAL
1: bash
5109 I loathe Harry Potter, Lord of the Rings and a...
5035 I am SOOOOOO sick of people claiming that Harr...
3945 i thought the da vinci code movie was really b...
5038 Wiccans react to possible Harry Potter book ba...
5050 Marcia Gaither, who teaches classes in Wiccani...
Name: Review, dtype: object
Enter comment to be analysed: fesadf
The comment is predicted Positive
(env) harrypotter0@harrypotter0-Lenovo-G50-80:~/Documents/final_year_project/backend/ml_model$
```

Figure: ML Model for Classification of comments

The objective of this investigation is to assess how YouTube videos are cited in academic

publications and to seek disciplinary differences in their use. In particular, the focus is on scholarly

uses of online videos by categorising the common topics of the videos cited in academic publications

across the sciences, medicine and health sciences, social sciences, and arts and

humanities. As

appropriate for the first study of its kind, the following questions address basic aspects of using online

videos in research communication.

1. How frequently are YouTube videos cited in academic publications? And has frequency of use

declined at any stage since the birth of YouTube (2005-2011)?

2. What types of YouTube videos are commonly cited in research articles?

3. Are there significant broad disciplinary differences in citing online videos?

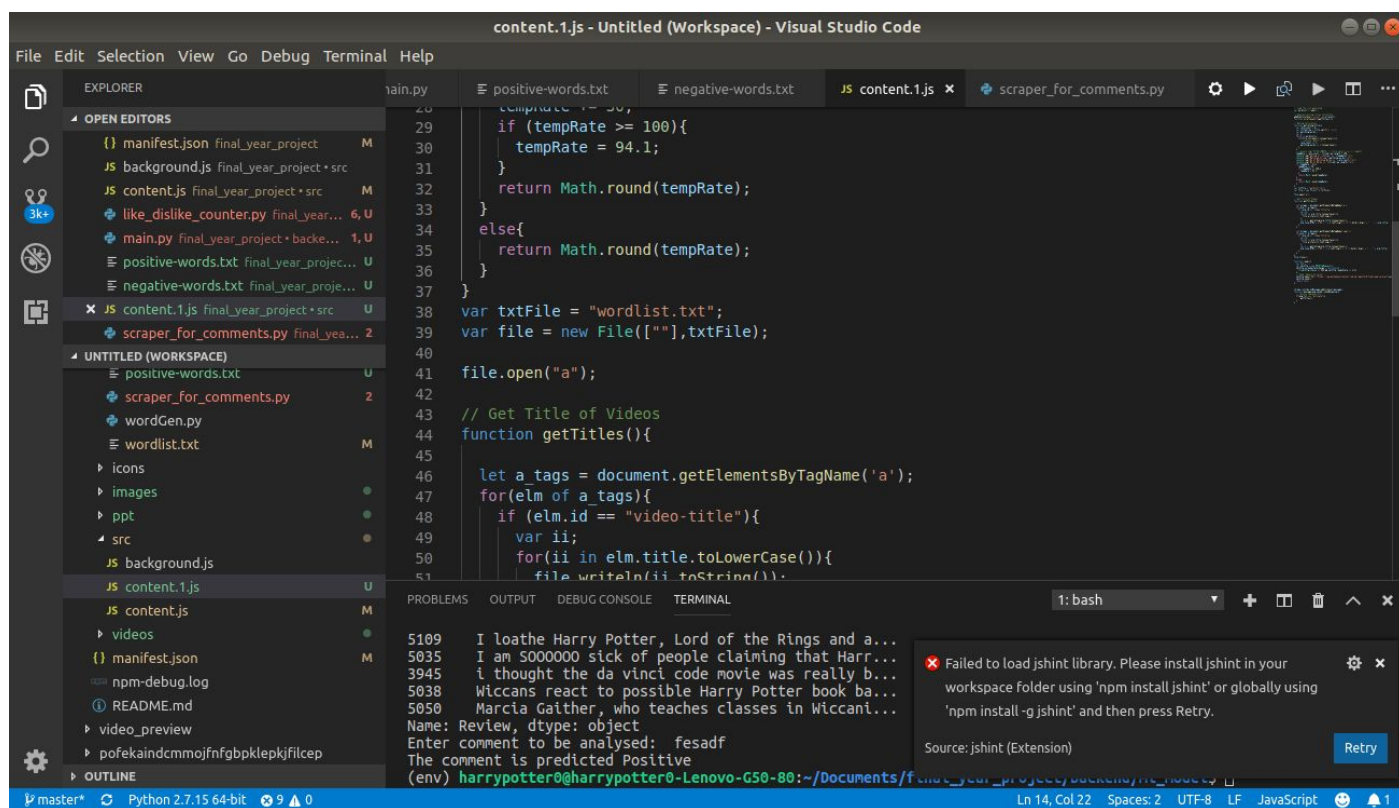


Figure: Script for handling the calls

Research in the area of mutli-label classification for rich video data has been limited in the past by the lack of an accurately labeled, large scale database of such videos. The release of the YouTube-8M dataset marks the beginning of a new set of opportunities to explore this nascent space. That said, the problem of multi label video classification can be compared to image labelling and image

captioning, both of which have been heavily studied in recent years. Investigation into the problem of multi-label image classification has taken off in recent years due to the creation of the ImageNet [3] database. Wang et al [4] created a hybrid CNN-RNN architecture that attempted to learn a joint image-label embedding that could be used to generate a set of distinct labels for a given image, as well as learn the semantic label dependency structure. Their approach provided the intuition behind our own approach to the problem of video classification, as did their decision to use beam search during inference. Hu et al [5] also applied a CNN-RNN architecture to the problem of multi-label image classification, but instead of using word embeddings as input to the recurrent sub-network, they instead use a word similarity matrix constructed from the WordNet [6] taxonomy. One limitation of Hu et al’s approach is that it relies heavily on the labels having a pre-existing, rich hierarchy of attributes that can be used to generate a structure from “coarse” attributes to “fine” labels. The Google Knowledge Graph taxonomy, in contrast to WordNet, uses Schema.org 5 entities, which lack rich attribute associations. Liu et al [7] expanded upon the work done by Wang et al [4] but separated the problem of learning the visual concepts (tags) from learning the concept similarity structure. They proposed using a semantic regularization embedding between the CNN image features and the RNN label features. The concept of separating this two subproblems for more efficient training is interesting, but there wasn’t sufficient time in this study to explore this idea fully, but we certainly believe it could potentially be used to improve training performance on our model. There has also been work done directly on the problem of multi-label video classification, but often with the addition of certain metadata features or raw visual/audio signals we didn’t have access to in this study. Yang and Toderici [8] combined metadata with raw video, but their metadata consisted of per user watching statistics, which we lacked in the YouTube-8M dataset. Jang et al [9] proposed a model called rDNN (Regularized Deep Neural Network) that extracts visual, audio, and trajectory features for each video and combines them using a series of deep fully connected layers. The label space is associated by concatenating label-level predictions into another series of fully connected layers for final prediction. In our work, we do something similar to fuse LSTM outputs together for final prediction.

TOOLS AND TECHNOLOGY USED

Tool and Libraries:

Gensim : <https://radimrehurek.com/gensim/>- **Gensim** is a robust open-source vector space modeling and topic modeling toolkit implemented in Python. It uses NumPy, SciPy and optionally Cython for performance.

Fasttext : <https://fasttext.cc/>-fasttext is a library for learning of word embeddings and text classification created by Facebook's AI Research (FAIR) lab.

NLTK : <https://www.nltk.org/>-

Technology:

Chrome Extension

Vanilla JavaScript

Tensorflow : As a base for Keras and more optimization.

Keras : For making Deep Learning Models

Pandas : For cleaning the data

Numpy : For mathematical purposes.

Plotly : For visualising graphs.

Scikit : For Shallow learning Algorithms.

Open CV : For computer vision part.

Convnet : For Handwritten Notes detection.

Pix2pix : Extension for video analysis.

Big Huge Labs API

XML parsing

CONCLUSION

Our effort has been to make it easier for improving the quality content available on You-Tube and thus enhancing the user-experience of the You-Tube users. They can easily get relevant information from the videos. This has never been approached before as previously people have tried to make recommendation system for these platforms rather than doing something about the quality of content available over there.

Also, the approach involves further advancements like usage of Deep NLP and Computer Vision for understanding the sentiments of the comments and also by analyzing the video itself frame-by-frame. So, this is a small effort from our side to improve the quality of videos and rank videos on You-Tube by providing them scores, lot of advancements will be seen in this area in the near future.

REFERENCES

Research Papers involved are :

- [1]. Green BF, Wolf AK, Chomsky C, and Laughery K. Baseball: An automatic question answerer.
- [2]. Weizenbaum J. ELIZA - a computer program for the study of natural language communication between man and machine.
- [3]. Woods W. Progress in Natural Language Understanding - An Application to Lunar Geology.
- [4]. Bobrow DG, Kaplan RM, Kay M, Norman DA, Thompson H, and Winograd T. Gus, a frame-driven dialog system.
- [5]. Katz B. Annotating the World Wide Web using natural language.
- [6]. Clark P, Thompson J, and Porter B. A knowledge-based approach to question answering.
- [7]. Riloff E and Thelen M. A Rule-based Question Answering System for Reading Comprehension Tests.
- [8]. Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems, Vol. 6, 2000, pp. 13-19.
- [9]. Ittycheriah A, Franz M, Zhu WJ, Ratnaparkhi A and Mammone RJ. IBM's statistical question answering system.