

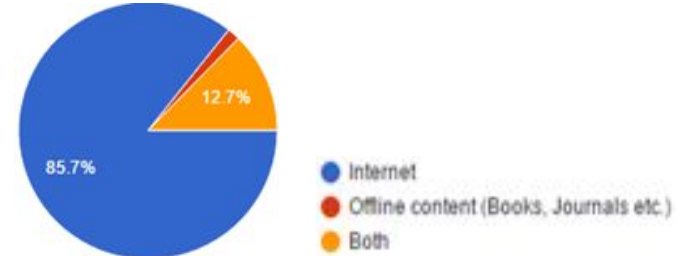
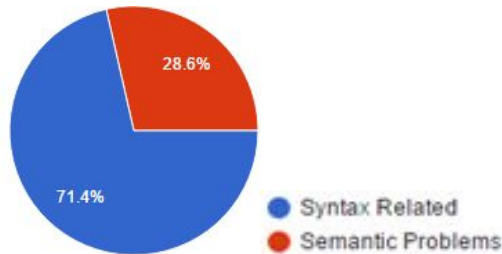
# PyReco

An Offline Stack Overflow Q&A  
Recommender

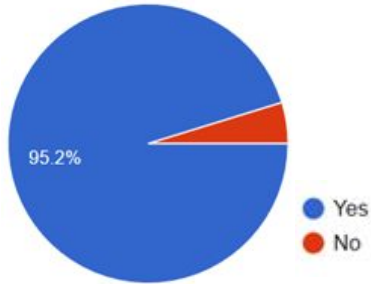
Ankit Kumar | Ashutosh Chaturvedi | Ayush Gupta | Harshdeep Kaur  
*{akumar18, achatu, agupta25, hkaur4}@ncsu.edu*

# Problem Statement

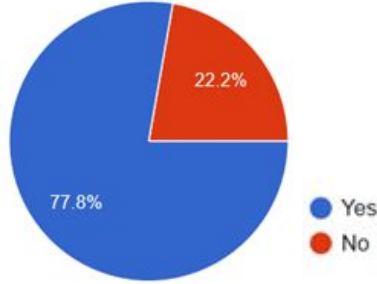
- **Syntax Issues**- A common programming problem which almost all programmers face.
- Finding Solutions?
  - Online: Search, specially on websites like stackoverflow.com (Major)
    - Drawbacks? Need for a good internet connection.
  - Offline: Books, etc (Minor)
    - Drawbacks? Difficult to obtain (cost and availability)
- User Survey: Based on the above findings, we conducted a user survey
- Results:



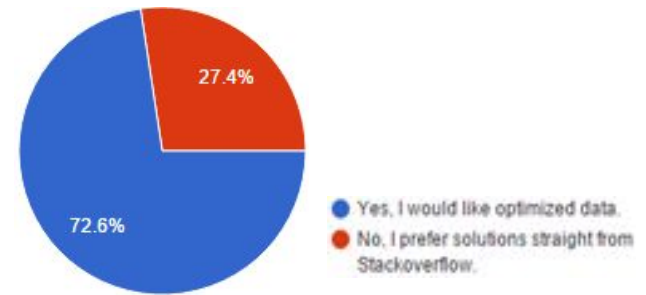
# Problem Statement (Cont.)



Percentage of users using StackOverflow.com to solve their problems



Percentage of programmers in favor of offline StackOverflow.com



Percentage of Programmers who would like optimized results over simple stackoverflow search

- What does the user need?
  - An offline solution
  - Better search results (independent of who is using)
  - Easy tool to enable quick and efficient debugging!

# Idea Formed

## **Idea:**

Design and implement a simple tool, which provides good search results to any user and at the same time does all this work offline!

## **Where do answers come from?**

The most popular search engine for programmers... Stackoverflow.com!

## **Simple Tool?**

Easy select keyword, and click to search using a Sublime Text plugin!

## **Offline and Portable?**

Host a local server using the MEAN stack and display results in a browser.

# Solutions Developed

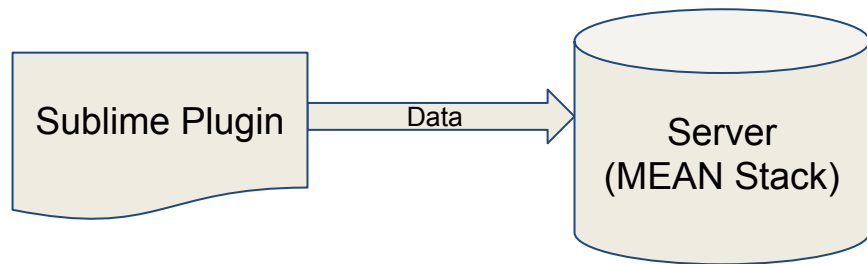
Approach: Build the simplest solution... Aim for a better solution, through improvements and additions to existing solution.

3 Solutions:

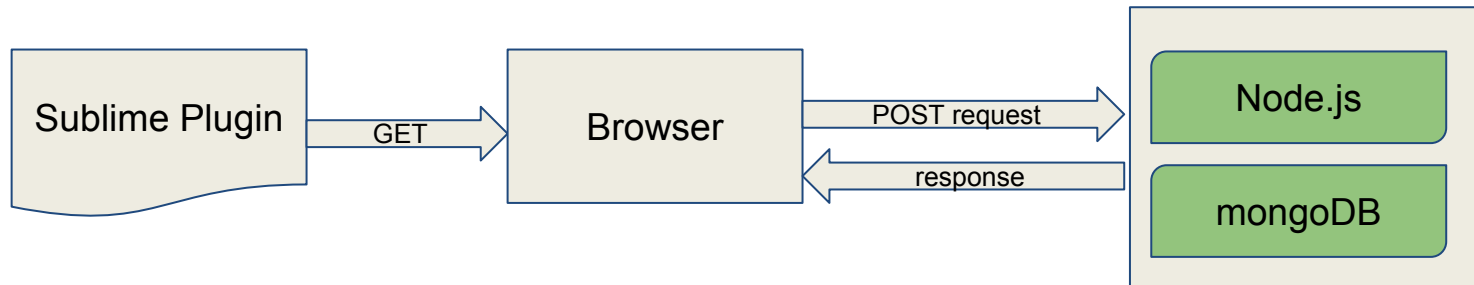
- **Bag of Words**
  - Map bag of keywords to different posts. Enable search through keyword match.
- **Clustering**
  - Cluster posts on the basis of keywords. Search for clusters, add filters to sort the posts and pick the top 10.
- **Clustering with Context Matching**
  - Add contextual data matching to clustering, to provide results closer to what the user wants.

# Design of Solutions

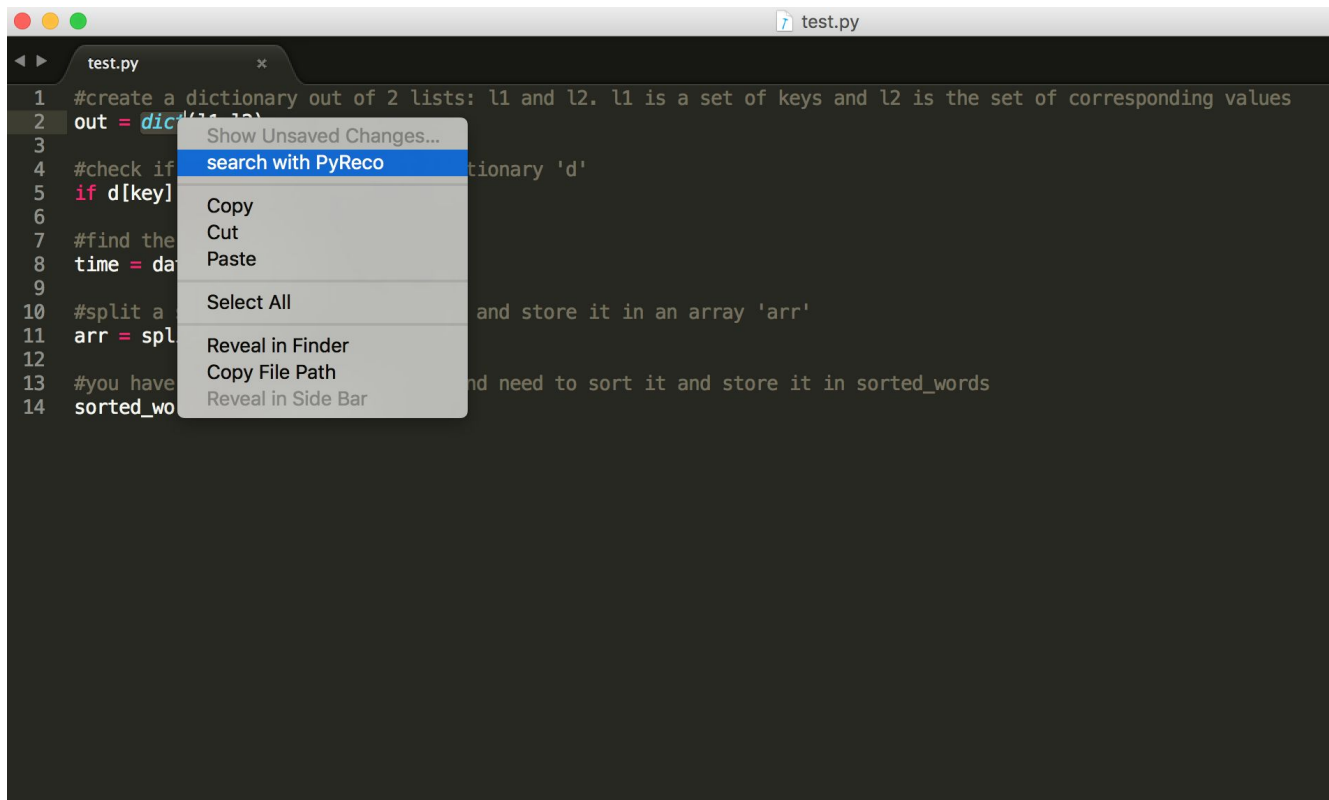
## Overview



## Implementation



# Sublime Plugin



# Solutions on Browser



## Welcome to PyReco! We make Python Easy!

### Your Search Results Are:

Title: Map two lists into a dictionary in Python

Body:

Imagine that you have: keys = ('name', 'age', 'food') values = ('Monty', 42, 'spam') What is the simplest way to produce the following dictionary ? dict = {'name' : 'Monty', 'age' : 42, 'food' : 'spam'} This code works, but I'm not really proud of it : dict = {} junk = map(lambda k, v: dict.update({k: v}), keys, values)

► Answer

Relevant



## Welcome to PyReco! We make Python Easy!

### Your Search Results Are:

Title: Map two lists into a dictionary in Python

Body:

Imagine that you have: keys = ('name', 'age', 'food') values = ('Monty', 42, 'spam') What is the simplest way to produce the following dictionary ? dict = {'name' : 'Monty', 'age' : 42, 'food' : 'spam'} This code works, but I'm not really proud of it : dict = {} junk = map(lambda k, v: dict.update({k: v}), keys, values)

▼ Answer

Like this: >>> keys = ['a', 'b', 'c'] >>> values = [1, 2, 3] >>> dictionary = dict(zip(keys, values)) >>> print dictionary {'a': 1, 'b': 2, 'c': 3} Voila :-)

The pairwise dict constructor and zip function are awesomely useful: <https://docs.python.org/2/library/functions.html#func-dict>

Relevant



# Bag of Words

- Sentence tokenization
  - Similar to StackOverflow
  - Stemming of features
- Stopwords removal
  - NLTK - 2400
  - Our own Stopwords - 950
- Keywords extraction
  - Bag of words
- Linear search
  - Keywords matching using MongoDB Text Indexing
  - Time consuming

# Clustering

- K-Means unsupervised learning
  - Easy to implement
  - Widely used for text based document clustering
- Data preprocessing
  - Tokenization
  - Vectorization - tf-idf vectors
- Model training
  - Cosine similarity
- Model testing
- Result is filtered on Vote Count, Number of Views, Answer rating

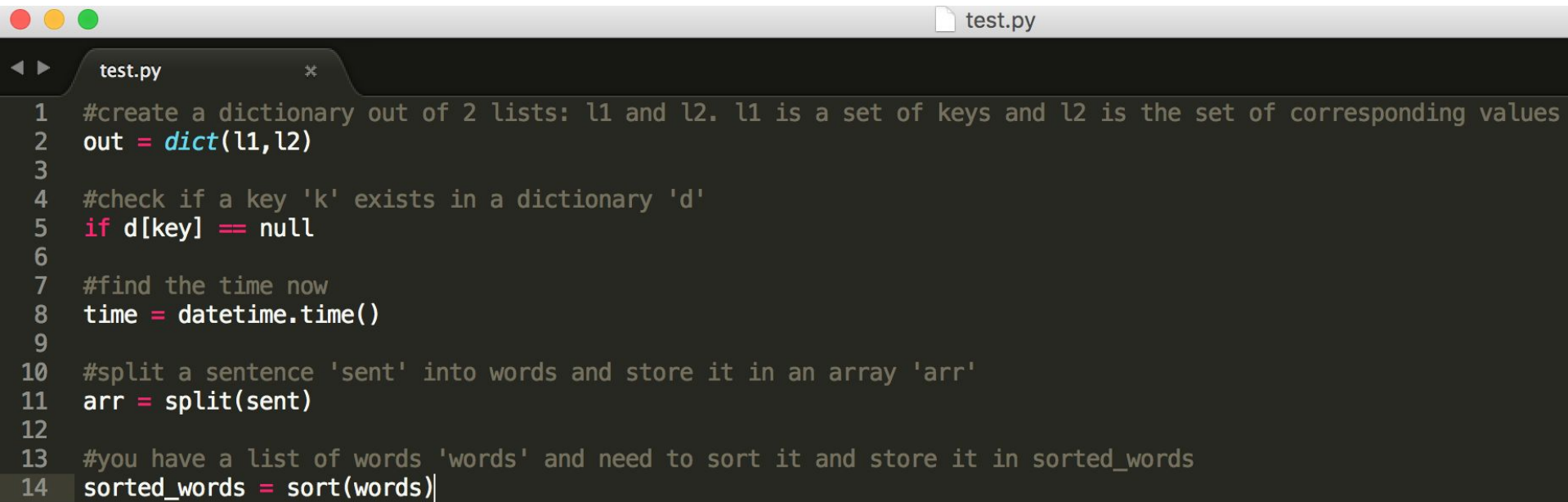
# Clustering with Context Matching

- Built on top of K-Means text clustering solution
- More Input data
  - Background contextual keywords from the user's code
  - Find posts closer w.r.t background contextual data
    - Precalculated SimHash fingerprints of posts
    - Compute SimHash of input contextual data
    - Select top few results based on minimization of Hamming Distance
- Context related result set
- Result is further filtered on Vote Count, Number of Views, Answer rating

# Setup for Testing

- Test: Edit a python code snippet using the tool provided
  - 5 lines of code, with one syntactical error in each line.
- Tool Provided: Our 3 Solutions and Stackoverflow.com.
- Participants: 20 (5 per solution)
- Data Collection for Evaluation
  - Telemetry: Statistical Evaluation
    - Answers Expanded
    - Clicks on the Button called “Relevant”
  - User Surveys: User’s Perspective
    - 5 questions for both quantitative and qualitative analysis
  - Bug Fixes: Usability in Syntactical Bug Fixing
    - Number of bugs fixed out of the 5 bugs given.

# Test Code



```
1 #create a dictionary out of 2 lists: l1 and l2. l1 is a set of keys and l2 is the set of corresponding values
2 out = dict(l1,l2)
3
4 #check if a key 'k' exists in a dictionary 'd'
5 if d[key] == null
6
7 #find the time now
8 time = datetime.time()
9
10 #split a sentence 'sent' into words and store it in an array 'arr'
11 arr = split(sent)
12
13 #you have a list of words 'words' and need to sort it and store it in sorted_words
14 sorted_words = sort(words)]
```

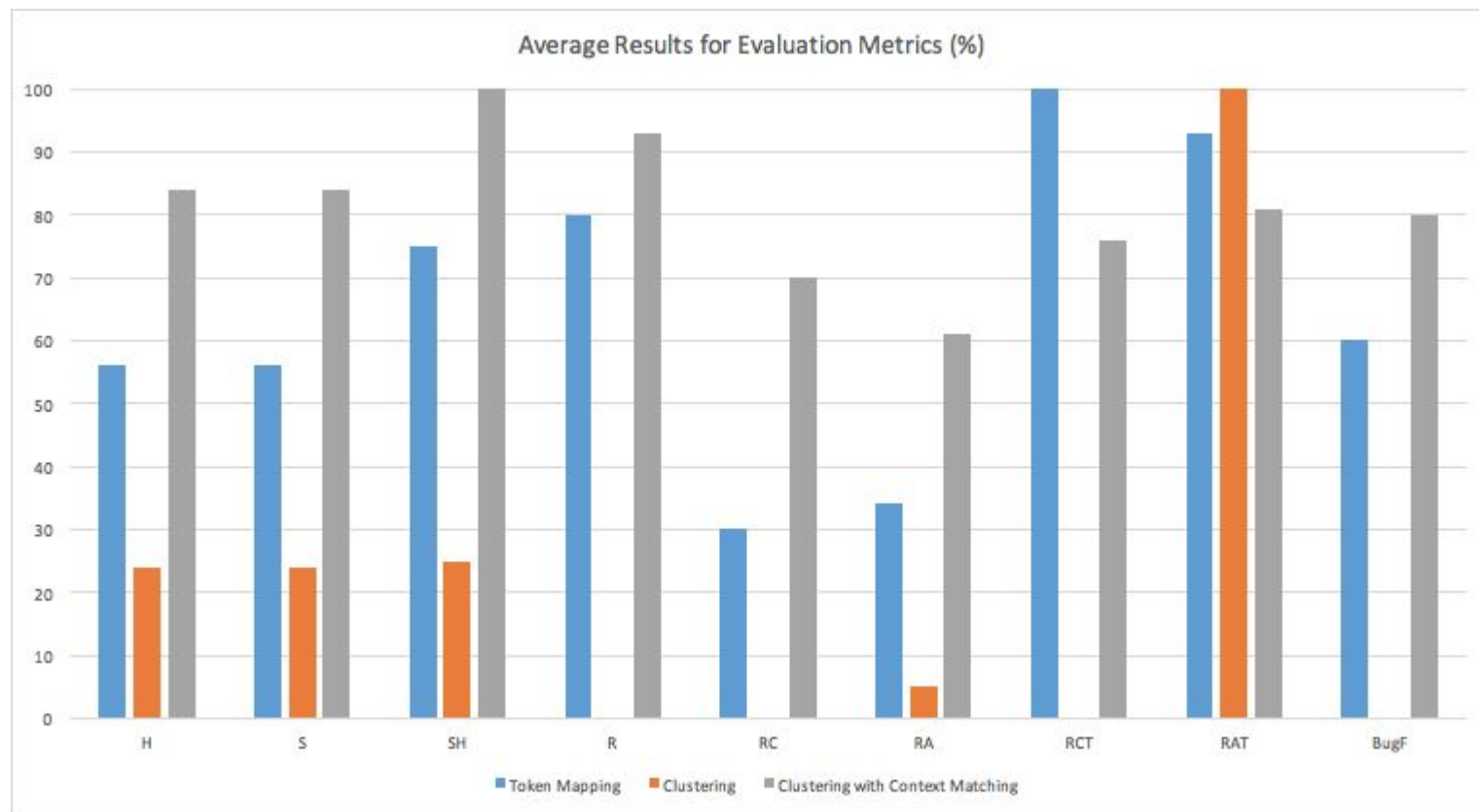
# Metrics for Evaluation

- All the metrics were averaged over all 5 users for every solution and were reported in percent values.
- User Surveys:
  - Helpfulness (H)
  - Overall Satisfaction (S)
  - Search Hits (SH)
  - Recommend? (R)
- Telemetry:
  - Ratio of Click on the Button “Relevant” for a solution to total number of clicks for the button (RC)
  - Ratio of Answers Expanded for a solution to total number of answers expanded (RA)

# Metrics for Evaluation (Cont.)

- Telemetry: (Cont.)
  - For a particular solution, the ratio of clicks on the button “Relevant” that for the top 5 search results, to the total number of clicks on the “Relevant” button for that solution.
  - For a particular solution, the ratio answers expanded for the top 5 search results, to the total number of answers expanded for that solution.
- Bug Fixes:
  - The ratio of bugs fixed to the total number of bugs. (BugF)

# Results





# Recommended Solution

- Clustering with Context Matching!
- Why?
  - Better evaluation results from telemetry.
    - Higher RA and RC, which suggests more relevant answers were available to the user.
  - Most number of bug fixes, which suggests better usability.
  - Better feedback from users
    - Higher S, SH, R and H which indicates better user satisfaction.

# Future Work

- Improve Keyword Search
  - Tf-idf based K-Means doesn't necessarily generate optimal models.
  - Experiment with other learning algorithms to generate better models and hence provide better recommendations.
- Provide user intervention
  - Our tool only allows for selecting keywords and finding solutions.
  - Provide users a way to search for keywords, not in the document that they are coding up, and with an option of not considering the background contextual data.
- Simplify Deployment
  - Deploying a MEAN server takes some time and effort.
  - Can use a simple SQLite DB with different types of interfaces
    - Command Line
    - Dedicated GUI

