Natural Language Processing (Automatic Questions Tagging System)

Student Name	ID	Department
مصطفي احمد احمد محمد	20201700822	CS
اية محمود محمد شحاتة	20201700175	CS
يوستينا وجدي وديع كامل	20201700992	CS
عبدالرحمن محيي محمد رمضان	20201700487	CS
احمد سامح دسوقي السيسي	20201700043	CS
احمد سعيد محمد النبوي	20201700046	CS

Contents

1 Introduction

- Project Overview
- Project Goals

2 Data Definition

- Dataset

3 Data Preprocessing

- Preprocessing on the Tags Dataset
- Preprocessing on the Questions Dataset

4 Feature Extraction

5 Data Splitting

6 Machine Learning Algorithm (Classification)

6.1 LinearSVC

6.1.1 LinearSVC Performance

- Grid Search
- Model Accuracy
- F1 Score

6.2 SGDClassifier

6.2.1 SGDClassifier Performance

- Model Accuracy
- F1 Score

6.3 Logistic Regression

6.3.1 Logistic Regression Performance
- Model Accuracy
- F1 Score

6.4 Decision Tree Classifier

6.4.1 Decision Tree Classifier Performance
- Model Accuracy
- F1 Score

7 Visualization
- Total Accuracy of all Models

8 Prediction

9 Conclusion

1

Introduction

1.1 Project Overview

The Automatic Question Tags System is a project that aims to assist users in generating question tags for their sentences with the help of Natural Language Processing (NLP) techniques. The system will be designed to analyze the grammatical structure of a given sentence and generate an appropriate question tag that fits the context and meaning of the sentence. The project will involve the use of machine learning algorithms to identify the relevant features in the sentences that can be used to generate the question tags. The system will also be trained on a large dataset of annotated sentences to improve its accuracy and effectiveness.

1.2 Project Goals

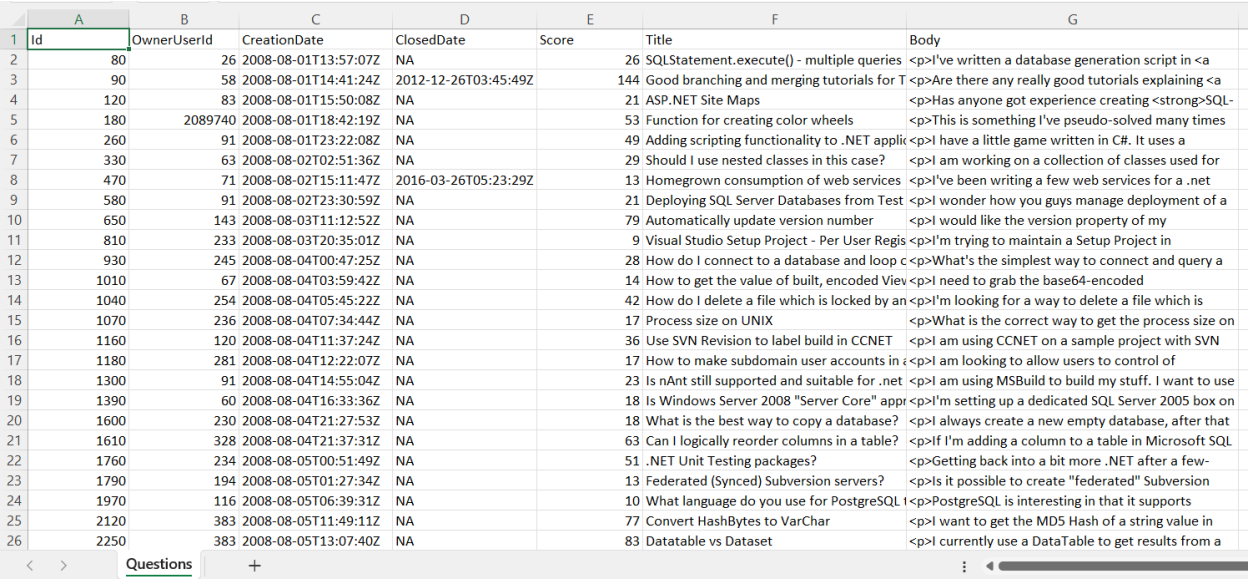
The primary goal of the Automatic Question Tags System project is to develop a tool that can automatically generate question tags for a given sentence. The system will be designed to accurately identify the relevant features in the sentence that can be used to generate the question tag, such as the verb tense, subject, and context. The system will also be trained on a large dataset of annotated sentences to improve its accuracy and effectiveness.

2

Data Definition

2.1 Dataset

- For our Questions dataset, we got a dataset containing 1264216 records of questions with 6 attributes for each of question, which will be explained in the next section.
    1. Dataset Sample:



	A	B	C	D	E	F	G
1	Id	OwnerUserId	CreationDate	ClosedDate	Score	Title	Body
2	80	26	2008-08-01T13:57:07Z	NA	26	SQLStatement.execute() - multiple queries	<p>I've written a database generation script in <a
3	90	58	2008-08-01T14:41:24Z	2012-12-26T03:45:49Z	144	Good branching and merging tutorials for T	<p>Are there any really good tutorials explaining <a
4	120	83	2008-08-01T15:50:08Z	NA	21	ASP.NET Site Maps	<p>Has anyone got experience creating SQL-
5	180	2089740	2008-08-01T18:42:19Z	NA	53	Function for creating color wheels	<p>This is something I've pseudo-solved many times
6	260	91	2008-08-01T23:22:08Z	NA	49	Adding scripting functionality to .NET appli	<p>I have a little game written in C#. It uses a
7	330	63	2008-08-02T02:51:36Z	NA	29	Should I use nested classes in this case?	<p>I am working on a collection of classes used for
8	470	71	2008-08-02T15:11:47Z	2016-03-26T05:23:29Z	13	Homegrown consumption of web services	<p>I've been writing a few web services for a .net
9	580	91	2008-08-02T23:30:59Z	NA	21	Deploying SQL Server Databases from Test	<p>I wonder how you guys manage deployment of a
10	650	143	2008-08-03T11:12:52Z	NA	79	Automatically update version number	<p>I would like the version property of my
11	810	233	2008-08-03T20:35:01Z	NA	9	Visual Studio Setup Project - Per User Regis	<p>I'm trying to maintain a Setup Project in
12	930	245	2008-08-04T00:47:25Z	NA	28	How do I connect to a database and loop c	<p>What's the simplest way to connect and query a
13	1010	67	2008-08-04T03:59:42Z	NA	14	How to get the value of built, encoded Viev	<p>I need to grab the base64-encoded
14	1040	254	2008-08-04T05:45:22Z	NA	42	How do I delete a file which is locked by an	<p>I'm looking for a way to delete a file which is
15	1070	236	2008-08-04T07:34:44Z	NA	17	Process size on UNIX	<p>What is the correct way to get the process size on
16	1160	120	2008-08-04T11:37:24Z	NA	36	Use SVN Revision to label build in CCNET	<p>I am using CCNET on a sample project with SVN
17	1180	281	2008-08-04T12:22:07Z	NA	17	How to make subdomain user accounts in a	<p>I am looking to allow users to control of
18	1300	91	2008-08-04T14:55:04Z	NA	23	Is nAnt still supported and suitable for .net	<p>I am using MSBuild to build my stuff. I want to use
19	1390	60	2008-08-04T16:33:36Z	NA	18	Is Windows Server 2008 "Server Core" appr	<p>I'm setting up a dedicated SQL Server 2005 box on
20	1600	230	2008-08-04T21:27:53Z	NA	18	What is the best way to copy a database?	<p>I always create a new empty database, after that
21	1610	328	2008-08-04T21:37:31Z	NA	63	Can I logically reorder columns in a table?	<p>If I'm adding a column to a table in Microsoft SQL
22	1760	234	2008-08-05T00:51:49Z	NA	51	.NET Unit Testing packages?	<p>Getting back into a bit more .NET after a few-
23	1790	194	2008-08-05T01:27:34Z	NA	13	Federated (Synced) Subversion servers?	<p>Is it possible to create "federated" Subversion
24	1970	116	2008-08-05T06:39:31Z	NA	10	What language do you use for PostgreSQL t	<p>PostgreSQL is interesting in that it supports
25	2120	383	2008-08-05T11:49:11Z	NA	77	Convert HashBytes to VarChar	<p>I want to get the MD5 Hash of a string value in
26	2250	383	2008-08-05T13:07:40Z	NA	83	Datatable vs Dataset	<p>I currently use a DataTable to get results from a

Questions

2. Dataset Info

```
RangeIndex: 1264216 entries, 0 to 1264215
Data columns (total 7 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   Id              1264216 non-null  int64
 1   OwnerUserId     1249762 non-null  float64
 2   CreationDate    1264216 non-null  object
 3   ClosedDate      55959 non-null    object
 4   Score           1264216 non-null  int64
 5   Title           1264216 non-null  object
 6   Body            1264216 non-null  object
dtypes: float64(1), int64(2), object(4)
memory usage: 67.5+ MB
None
```

- For our Tags dataset, we got a dataset containing 3750994 records of tags with one attribute for each tag, which will be explained in the next section.
    1. Dataset Sample:

	A	B
1	Id	Tag
2	80	flex
3	80	actionscript-3
4	80	air
5	90	svn
6	90	tortoisesvn
7	90	branch
8	90	branching-and-merging
9	120	sql
10	120	asp.net
11	120	sitemap
12	180	algorithm
13	180	language-agnostic
14	180	colors
15	180	color-space
16	260	c#
17	260	.net
18	260	scripting
19	260	compiler-construction
20	330	c++
21	330	oop
22	330	class
23	330	nested-class
24	470	.net
25	470	web-services
26	580	sql-server

Tags

    2. Dataset Info

```
RangeIndex: 3750994 entries, 0 to 3750993
Data columns (total 2 columns):
 #   Column  Dtype
---  ------  -----
 0   Id      int64
 1   Tag     object
dtypes: int64(1), object(1)
memory usage: 57.2+ MB
None
```

3

Data Preprocessing

3.1 Preprocessing of Tags:

the 'tags output' dataframe is converted to a string, and the frequency of each tag is calculated and stored in the 'tags_fre' dataframe. The top 50 most frequent tags are stored in the 'most_tags' dataframe.

	Tag frequency	Tag
c#	4657	c#
.net	2628	.net
java	2436	java
asp.net	2143	asp.net
php	1884	php
javascript	1799	javascript
c++	1686	c++
python	1329	python
jquery	1249	jquery
sql	1234	sql
iphone	1205	iphone
sql-server	1016	sql-server
html	909	html
mysql	847	mysql
asp.net-mvc	742	asp.net-mvc
c	716	c
ruby-on-rails	646	ruby-on-rails
wpf	642	wpf

The 'tags_output' dataframe is then merged with the 'tags_fre' dataframe, and only those tags with a frequency of at least 271 are retained. The resulting dataframe is stored in 'merge_tags'.

	Id	Tag	Tag frequency
0	80	flex	329
3	90	svn	291
7	120	sql	1234
8	120	asp.net	2143
10	180	algorithm	300
...	...	...	...
99985	1750010	c#	4657
99989	1750040	php	1884
99990	1750040	mysql	847
99993	1750070	c#	4657
99997	1750170	visual-studio	548

40449 rows × 3 columns

'grouped_tags' is then created by grouping the 'merge_tags' dataframe by 'Id' and concatenating the tags for each 'Id' into a single string. Finally, 'questions_tags' is created by merging the 'questions' dataframe with the 'grouped_tags' dataframe on the 'Id' column.

	Id	OwnerUserId	CreationDate	ClosedDate	Score	Title	Body	Tag	tag_count
0	80	26.0	2008-08-01T13:57:07Z	NaN	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...	[flex]	1
1	90	58.0	2008-08-01T14:41:24Z	2012-12-26T03:45:49Z	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...	[svn]	1
2	120	83.0	2008-08-01T15:50:08Z	NaN	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...	[sql, asp.net]	2
3	180	2089740.0	2008-08-01T18:42:19Z	NaN	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...	[algorithm]	1

3.2 Preprocessing of Questions:

The preprocessing of questions (Body column and Title) involves several techniques, including removing HTML tags, tokenization, stop word removal, lemmatization, and cleaning punctuation. We first removed HTML tags from the Body column only using the BeautifulSoup class from the bs4 library. We then tokenized the text using the ToktokTokenizer class from the nltk library. We also removed stop

words, punctuation, and lemmatized the text using the WordNetLemmatizer class from the nltk library, and this is results:

```
0    sqlstatement.execute( multiple query one state...
1            good branching merge tutorial tortoisesvn
2                                      asp.net site map
3                          function create color wheel
4            add script functionality .net application
Name: clean_title, dtype: object
```

```
0    write database generation script sql want exec...
1    really good tutorial explain branch merge apac...
2    anyone get experience create sql-based asp.net...
3    something pseudo-solved many time never quite ...
4    little game write c#. us database back-end. tr...
Name: clean_body, dtype: object
```

4

Feature Extraction

The feature extraction process in the Automatic Question Tags System involves using the TfidfVectorizer class from the sklearn library to convert the preprocessed text data into numerical vector representations. The vectorizer computes the Term Frequency - Inverse Document Frequency (TF-IDF) scores for each token in the text, which are then used as features. The system also limits the maximum number of features to 1,000 and uses n-grams of up to 3 words to capture the context of the text, The TfidfVectorizer class returns a sparse matrix of

shape (n_samples, n_features), where n_samples is the number of questions and n_features is the number of unique tokens in the text. Each element of the matrix represents the tf-idf weight of a token in a particular question. The resulting matrix is a numerical representation of the text data that can be used as input to a machine learning algorithm.

5

Data Splitting

We use " train test split" , The process of dividing the data into these sets is called a train-test split. The train-test split is typically done randomly, with a certain percentage of the data (80%) allocated to the training set and the remaining percentage (20%) allocated to the test set and apply preprocessing to train and test data.

6

Machine Learning Algorithm

Several machine learning algorithms are used to predict the tags for new questions, including SGDClassifier, LogisticRegression, LinearSVC, and DecisionTreeClassifier. The OneVsRestClassifier class from the sklearn library is used to train the multi-label classification model.

6.1 LinearSVC:

```
C=[1,2,3,4,5,6,7,8,9,10]
for i in C:
  svm_model = LinearSVC(C=i)
  clf = OneVsRestClassifier(svm_model)
  clf.fit(x_train, y_train)
  y_pred = clf.predict(x_test)
  print_score(y_pred, clf)
```

6.1.1 LinearSVC Performance

- Grid search: The provided code implements a grid search for the best C parameter in a linear SVM model using the scikit-learn library. The SVM model is trained using the LinearSVC class and the OneVsRestClassifier wrapper for multi-class classification. The C parameter controls the trade-off between maximizing the margin and minimizing the classification error of the SVM model, and is searched over a range of values using a for loop. The performance of the model is evaluated using a custom print_score function that computes and prints various performance metrics for each value of C. The grid search approach provides a systematic and data-driven way to select the best value of C that maximizes the performance of the model on the test set. And this is output:

```
Clf:  OneVsRestClassifier
F1 Score: 0.511294490289239
accau Score: 0.27888300784814746
----
Clf:  OneVsRestClassifier
F1 Score: 0.5156300879192445
accau Score: 0.27997809819310093
----
Clf:  OneVsRestClassifier
F1 Score: 0.5154523618895116
accau Score: 0.2772403723307173
----
Clf:  OneVsRestClassifier
F1 Score: 0.5145746109487321
accau Score: 0.27505019164081035
----
Clf:  OneVsRestClassifier
F1 Score: 0.5125195618153364
accau Score: 0.27395510129585693
----
Clf:  OneVsRestClassifier
F1 Score: 0.508918876970528
accau Score: 0.2697572549735353
----
Clf:  OneVsRestClassifier
F1 Score: 0.5073043210825773
accau Score: 0.26902719474356634
----
Clf:  OneVsRestClassifier
F1 Score: 0.5055415424596804
accau Score: 0.26738455922613613
----
Clf:  OneVsRestClassifier
F1 Score: 0.5043385599025727
accau Score: 0.265011863478737
```

- Accuracy of LinearSCV: 0.27997809819310093

- F1Score of LinearSCV: 0.5156300879192445

6.2 SGDClassifier:

The SGDClassifier is a linear model that trains on a subset of the data at each iteration using stochastic gradient descent. It is well-suited for large-scale and sparse datasets, and can handle a variety of loss functions and penalties. The SGDClassifier is fast and efficient, but may require careful tuning of the learning rate and regularization parameters to achieve good performance.

6.2.1 SGDClassifier Performance

- Accuracy of SGDClassifier: 0.2463953276145282
- F1Score of SGDClassifier: 0.45944729498604736

6.3 Logistic Regression:

Logistic Regression is a linear model that estimates the probability of a binary or multi-class outcome using a logistic or softmax function. It is a simple and interpretable model that can handle both continuous and categorical features. Logistic Regression is widely used for binary classification and can be extended to multi-class classification using the one-vs-rest or multinomial approach. Logistic Regression is a well-established model with a strong theoretical foundation and can achieve high performance on a variety of datasets.

6.3.1 Logistic Regression Performance

- accuracy of Logistic Regression: 0.225777112611790473
- the F1 Score of Logistic Regression: 0.4390684239032316 4

6.4 Decision Tree Classifier:

The DecisionTreeClassifier is a tree-based model that constructs a decision tree to recursively partition the data into different classes based on the values of the features. It is a simple and interpretable model that can handle both continuous and categorical features, and can capture non-linear and interaction effects. The DecisionTreeClassifier can be prone to overfitting and may require careful tuning of the hyperparameters, such as the maximum depth, minimum number of samples per leaf, and minimum number of samples per split, to balance between simplicity and complexity and achieve good performance on the test set.

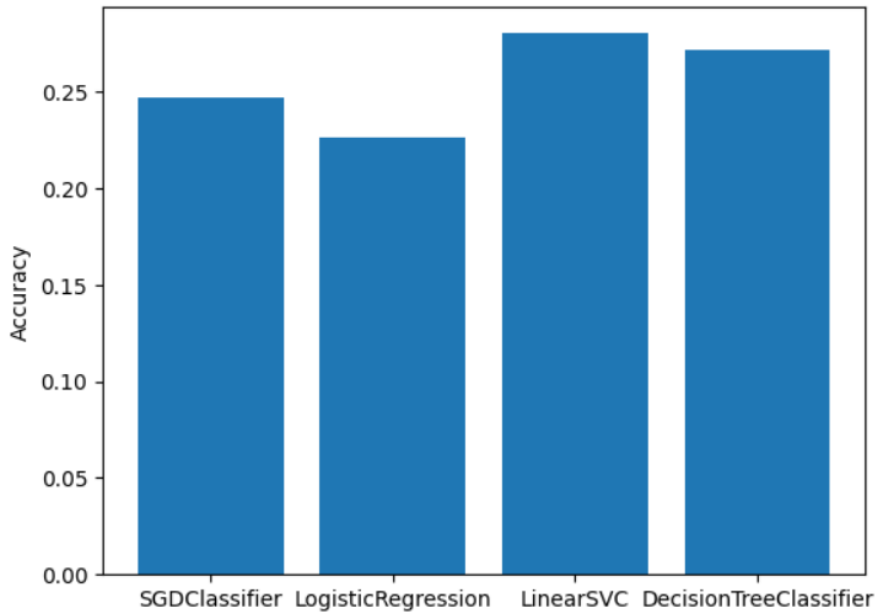
6.4.1 Decision Tree Classifier Performance

- accuracy of DecisionTreeClassifier: 0.27194743566344226
- the F1 Score of DecisionTreeClassifier: 0.5215875782302147

7

Visualization

Create a bar plot of the accuracy scores for four different classification models evaluated on a given dataset. The plot visualizes the performance of each model and provides a quick and easy way to compare the accuracy scores of the models. The plot can be customized and extended to include additional information and provide more insights into the performance of the models. The visualization can help in selecting the best model for a given task and assessing the performance of the model on the test set.



8

Prediction

This code shows a simple and effective way to predict the tags of a given text input using machine learning and text preprocessing techniques. The code takes a text input 'how to write ml code in python and c#' and preprocesses it by removing stop words and punctuations, and applying lemmatization. The code then converts the preprocessed text into a feature vector using a TF-IDF vectorizer object, and applies a trained classifier object to predict the label of the text input.

```
x = 'how to write ml code in python and c#'
x=[token_remove_stop_words_punc_lemma(x)]
print(x)
xt = tidf_X1.transform(x)
s = clf.predict(xt)
```

This is the output :

```
['write ml code python c#']
```

```
m.inverse_transform(s)
```

```
[('c#', 'python')]
```

9

Conclusion

The Automatic Question Tags System project is an innovative application of NLP techniques that aims to assist users in generating question tags for their sentences. The system will use machine learning algorithms to identify the relevant features in the sentences and improve its accuracy over time. The project has many potential applications in language learning, content creation, and social media, and has the potential to improve communication and language skills for users across a variety of domains.