

Predicting the Programming Language of Questions and Snippets of StackOverflow Using Natural Language Processing

Kamel Alreshedy, Dhanush Dharmaretnam, Daniel M. German, Venkatesh Srinivasan and T. Aaron Gulliver

Department of Computer Science, University of Victoria

Kamel, Dhanushd, dmg, srinivas@uvic.ca, agullive@ece.uvic.ca

Presented by

Turzo Ahsan Sami (ID: 20166012)

In partial fulfillment of the module
CSE-712: Natural Language Processing



Contents

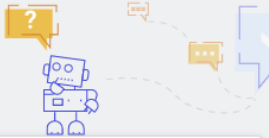
- Introduction
- Related Work
- Research Questions
- Dataset Extraction & Processing
- Methodology
- Results
- Shortcomings
- Conclusion & Future Work
- What I have learned reading this paper

Introduction

- Stack Overflow:
 - The most popular Q/A website among programmers & software developers
 - Questions usually contain text body and code snippets
 - Relies on users to properly tag the programming language of a question
 - Groups questions based on tags
 - Chances of human error : New users may not tag their posts correctly
 - Without proper tags posts get downvoted and flagged by moderators
 - Questions related to frameworks and libraries may not have proper tags
 - Pandas is a Python library: it's questions usually do not include a "Python" tag
 - Angular 2+ uses Typescript however it's questions rarely have "Typescript" tag

Introduction (continued)











  Products



Ask a public question

Title
Be specific and imagine you're asking a question to another person

Body
Include all the information someone would need to answer your question

B *I*          

Links Images Styling/Headers Lists Blockquotes Code HTML [More](#)

`code`

bold

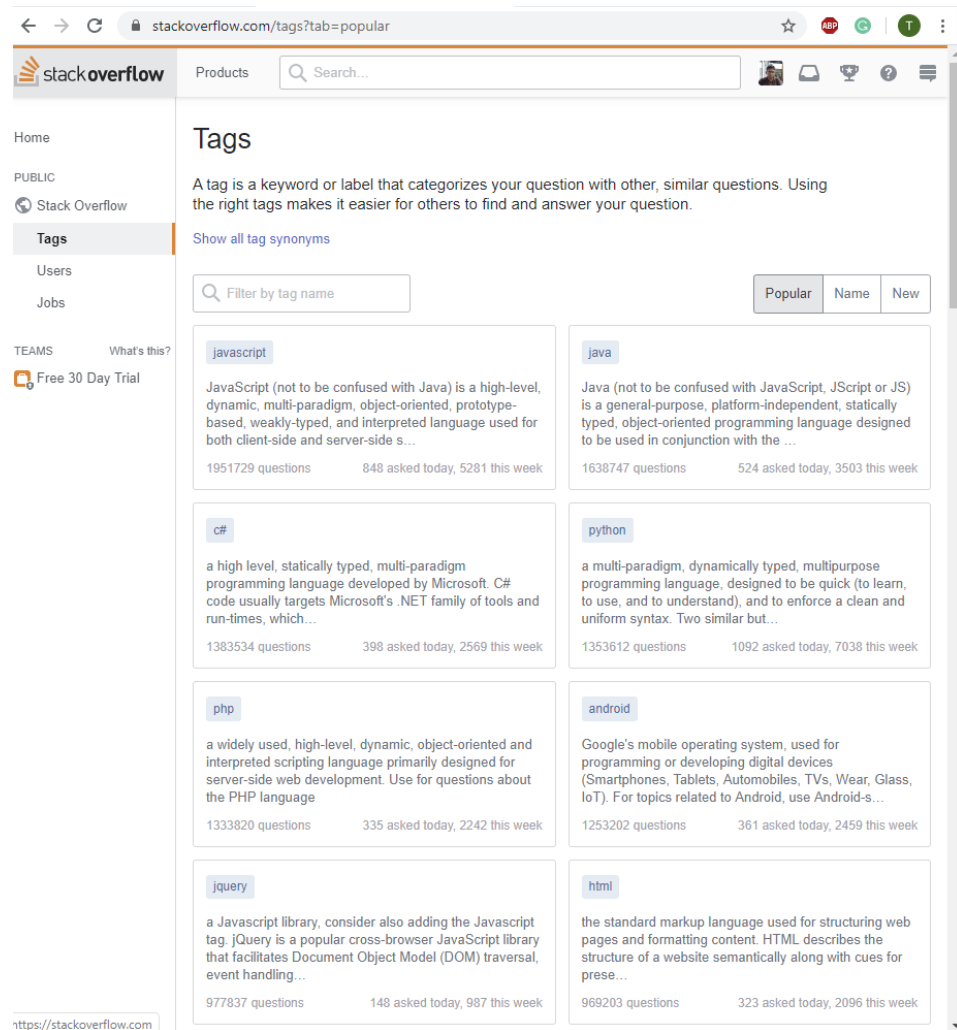
italic

>quote

Tags
Add up to 5 tags to describe what your question is about

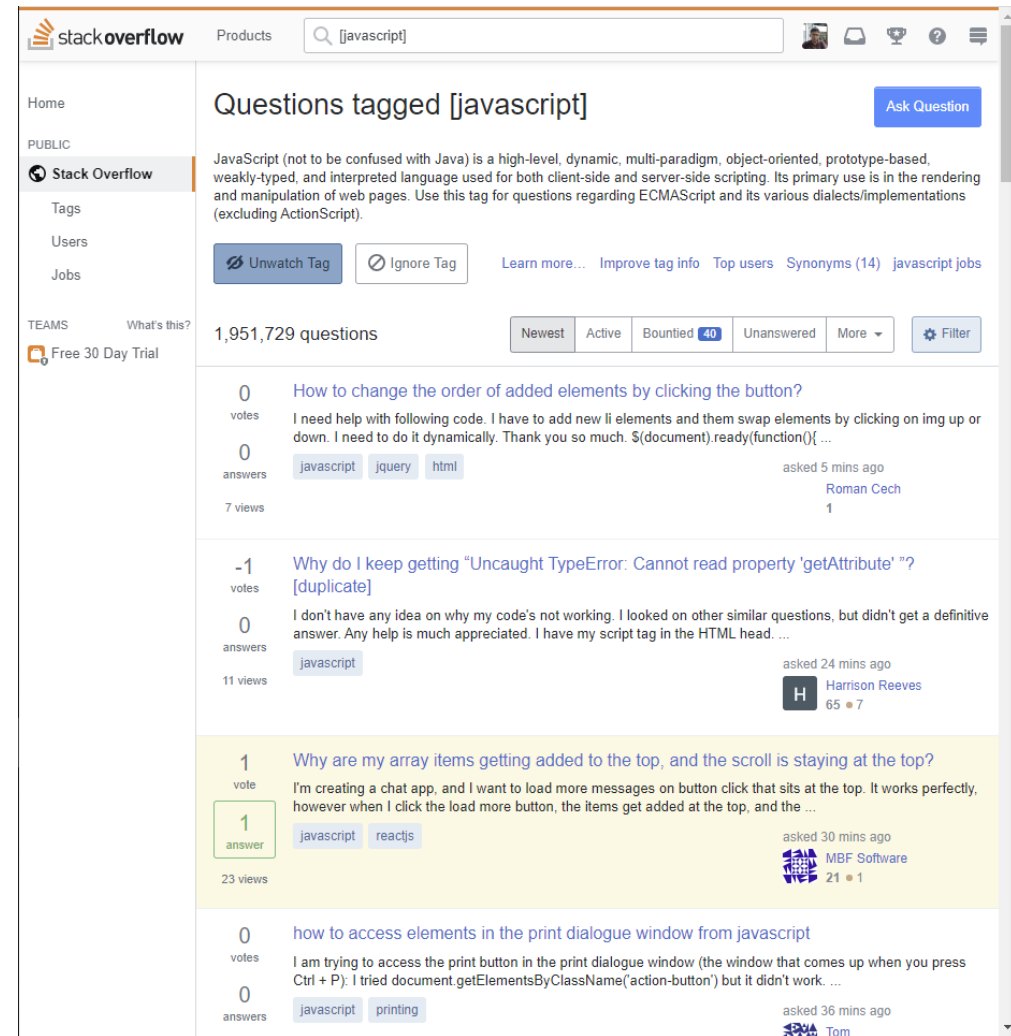
Review your question

Introduction (continued)



The screenshot shows the Stack Overflow 'Tags' page. The left sidebar contains navigation links: Home, PUBLIC (Stack Overflow), Tags (selected), Users, Jobs, TEAMS, and What's this? (Free 30 Day Trial). The main content area is titled 'Tags' and includes a description: 'A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.' Below this is a search bar 'Filter by tag name' and buttons for 'Popular', 'Name', and 'New'. A grid of tag cards is displayed, each with a tag name, a description, and statistics (questions, asked today, asked this week). The tags shown are: javascript, java, c#, python, php, android, jquery, and html.

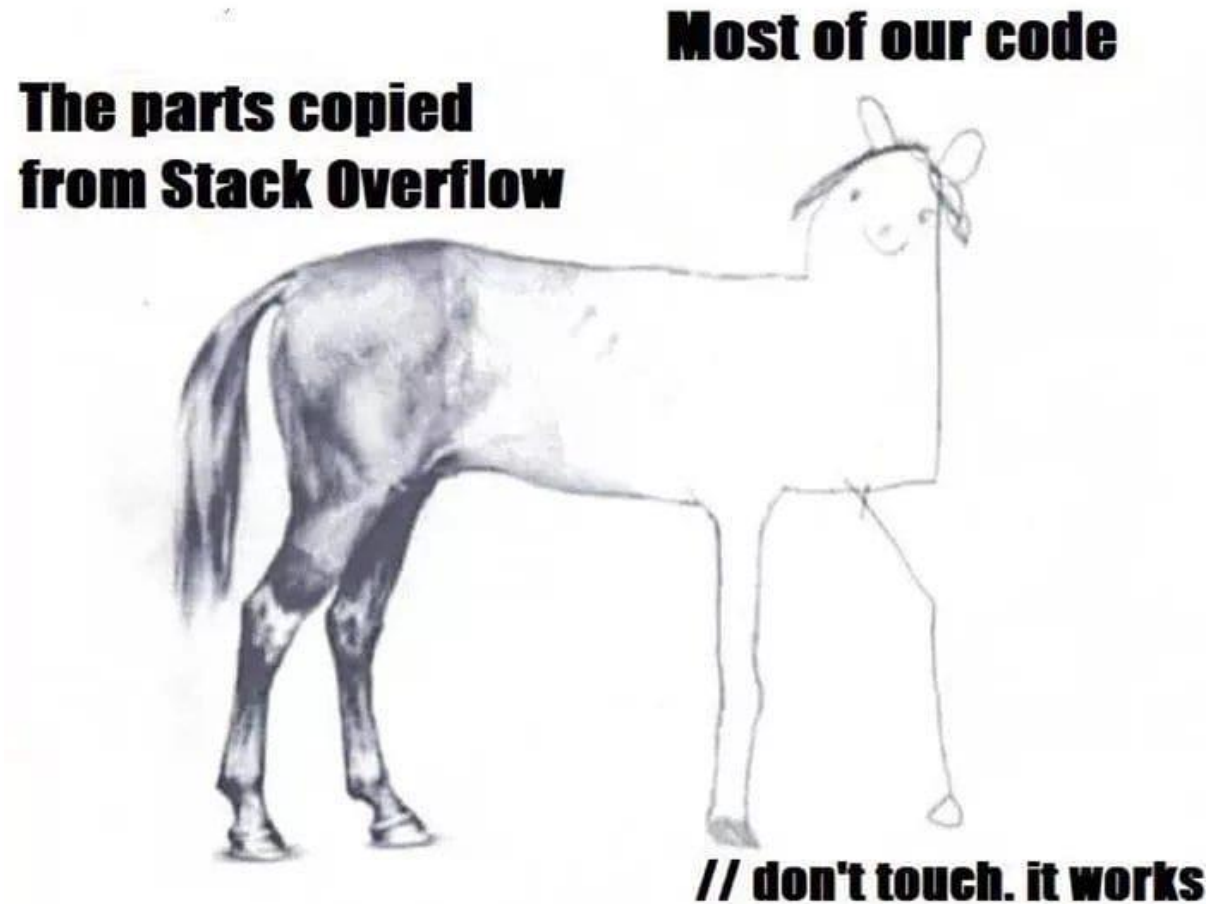
Tag	Description	Questions	Asked Today	Asked This Week
javascript	JavaScript (not to be confused with Java) is a high-level, dynamic, multi-paradigm, object-oriented, prototype-based, weakly-typed, and interpreted language used for both client-side and server-side s...	1951729	848	5281
java	Java (not to be confused with JavaScript, JScript or JS) is a general-purpose, platform-independent, statically typed, object-oriented programming language designed to be used in conjunction with the ...	1638747	524	3503
c#	a high level, statically typed, multi-paradigm programming language developed by Microsoft. C# code usually targets Microsoft's .NET family of tools and run-times, which...	1383534	398	2569
python	a multi-paradigm, dynamically typed, multipurpose programming language, designed to be quick (to learn, to use, and to understand), and to enforce a clean and uniform syntax. Two similar but...	1353612	1092	7038
php	a widely used, high-level, dynamic, object-oriented and interpreted scripting language primarily designed for server-side web development. Use for questions about the PHP language	1333820	335	2242
android	Google's mobile operating system, used for programming or developing digital devices (Smartphones, Tablets, Automobiles, TVs, Wear, Glass, IoT). For topics related to Android, use Android-s...	1253202	361	2459
jquery	a Javascript library, consider also adding the Javascript tag. JQuery is a popular cross-browser JavaScript library that facilitates Document Object Model (DOM) traversal, event handling...	977837	148	987
html	the standard markup language used for structuring web pages and formatting content. HTML describes the structure of a website semantically along with cues for prese...	969203	323	2096



The screenshot shows the Stack Overflow 'Questions tagged [javascript]' page. The left sidebar contains navigation links: Home, PUBLIC (Stack Overflow), Tags (selected), Users, Jobs, TEAMS, and What's this? (Free 30 Day Trial). The main content area is titled 'Questions tagged [javascript]' and includes a description: 'JavaScript (not to be confused with Java) is a high-level, dynamic, multi-paradigm, object-oriented, prototype-based, weakly-typed, and interpreted language used for both client-side and server-side scripting. Its primary use is in the rendering and manipulation of web pages. Use this tag for questions regarding ECMAScript and its various dialects/implementations (excluding ActionScript).' Below this are buttons for 'Unwatch Tag' and 'Ignore Tag', and links for 'Learn more...', 'Improve tag info', 'Top users', 'Synonyms (14)', and 'javascript jobs'. A filter bar shows '1,951,729 questions' and filters for 'Newest', 'Active', 'Bountied 40', 'Unanswered', and 'More'. A list of questions is displayed, each with a title, votes, answers, tags, and a brief description.

Question Title	Votes	Answers	Tags	Asked	Asked By
How to change the order of added elements by clicking the button?	0	0	javascript, jquery, html	asked 5 mins ago	Roman Cech
Why do I keep getting "Uncaught TypeError: Cannot read property 'getAttribute' "? [duplicate]	-1	0	javascript	asked 24 mins ago	Harrison Reeves
Why are my array items getting added to the top, and the scroll is staying at the top?	1	1	javascript, reactjs	asked 30 mins ago	MBF Software
how to access elements in the print dialogue window from javascript	0	0	javascript, printing	asked 36 mins ago	Tom

Introduction (continued)



Introduction (continued)

- This paper proposes a classifier to predict programming language of questions asked on Stack Overflow using NLP and ML.
- The classifier achieves an accuracy of 91.1% in predicting the 24 most popular programming languages.
- Combined features from the title, body and the code snippets of the question.
- The results demonstrate that it is possible to identify the programming language of a snippet of few lines of source code.

Related Works

- J. F. Baquero, J. E. Camargo, F. Restrepo-Calle, J. H. Aponte, and F. A. González,
“Predicting the Programming Language: Extracting Knowledge from Stack Overflow Posts” (2017)
 - Extracted a set of 18,000 questions from Stack Overflow.
 - Dataset contained both text and code snippets.
 - Trained two classifiers using a Support Vector Machine model.
 - Achieved an accuracy of 60% for text body features and 44% for code snippets.
- J. Kennedy, V. Dam and V. Zaytsev,
“Software Language Identification with Natural Language Classifiers” (2016)
 - Identify the programming language of entire source code files from GitHub.
 - Classifier is based on five statistical language models from NLP.
 - Identifies 19 programming languages, with accuracy of 97.5%.

Related Works (continued)

- J. N. Khasnabish, M. Sodhi, J. Deshmukh, and G. Srinivasaraghavan,
“Detecting Programming Language from Source Code Using Bayesian Learning Techniques” (2014)
 - Proposed a model to detect 10 programming languages using source code files.
 - Achieved an accuracy of 93.48% using Multinomial Naive Bayes.
- V. S. Rekha, N. Divya, and P. S. Bagavathi,
“A Hybrid Autotagging System for StackOverflow Forum Questions” (2014)
 - Proposed a hybrid auto-tagging system that suggests tags to users proposed a hybrid auto-tagging system that suggests tags to users.
 - Multinomial Naive Bayes (MNB) was trained and tested for the proposed classifier which achieved 72% accuracy.

Research Questions

- RQ1. Can we predict the programming language of a question in Stack Overflow?
- RQ2. Can we predict the programming language of a question in Stack Overflow without using code snippets inside it?
- RQ3. Can we predict the programming language of code snippets in Stack Overflow questions?

Research Questions (continued)

- RQ1. Can we predict the programming language of a question in Stack Overflow?
 - Evaluate how machine learning performs when all the information in a Stack Overflow question is used.
 - Includes question's title, body (textual information) and code snippets in it.

Research Questions (continued)

- RQ2. Can we predict the programming language of a question in Stack Overflow without using code snippets inside it?
 - Determine whether the inclusion of code snippets is an essential factor to determine the programming language that a question refers to.
 - Includes textual information only, omitting the question title and code snippets.

Research Questions (continued)

- RQ3. Can we predict the programming language of code snippets in Stack Overflow questions?
 - Evaluate the ability to use machine learning to predict the language of a snippet of source code.
 - Includes code snippets only.

Dataset Extraction & Processing

- The Stack Overflow July 2017 data dump was used for analysis.
- 37.21 million posts, of which 14.45 million are questions with 50.9k different tags.
- The most popular 24 programming languages as per the 2017 Stack Overflow developer survey were selected.
- Constitute about 93% of the questions.
- Selected languages: Assembly, C, C#, C++, CoffeeScript, Go, Groovy, Haskell, Java, JavaScript, Lua, Matlab, Objective-c, Perl, PHP, Python, R, Ruby, Scala, SQL, Swift, TypeScript, Vb.Net, Vba.

Dataset Extraction & Processing (continued)

- Questions with more than one programming language tags were removed
- Questions chosen contained at least one code snippet, and the code snippet had at least 10 characters
- For each programming languages: approximately 10,000 random questions
- The total number of questions selected was 232,727
- XMLTODICT: parsed .xml data
- Python BeautifulSoup: extract code and text separately from questions

Dataset Extraction & Processing (continued)

- ML models cannot be trained on raw text because of noise present.
- The textual information need to be preprocessed.
- Preprocessing steps:
 - non-alphanumeric characters such as *punctuation, numbers and symbols* were removed
 - entity names were identified using the Spacy Library (<https://spacy.io/>)
 - stop words such as *after, about, all, and, from* etc. were removed
 - stemming and lemmatization- using the NLTK library in Python

Dataset Extraction & Processing (continued)

- Dataset was split using the Tf-IDF vectorizer from the Scikit-learn library.
- The Minimum Document Frequency (min-df) was set to 10.
 - only words present in at least ten documents were selected
- Eliminates infrequent words from the dataset
 - ML models learn from the most important vocabulary
- The Maximum Document Frequency (max-df) was set to default.
- The datasets were split into training and test data using the ratio 80:20.

Dataset Extraction & Processing (continued)

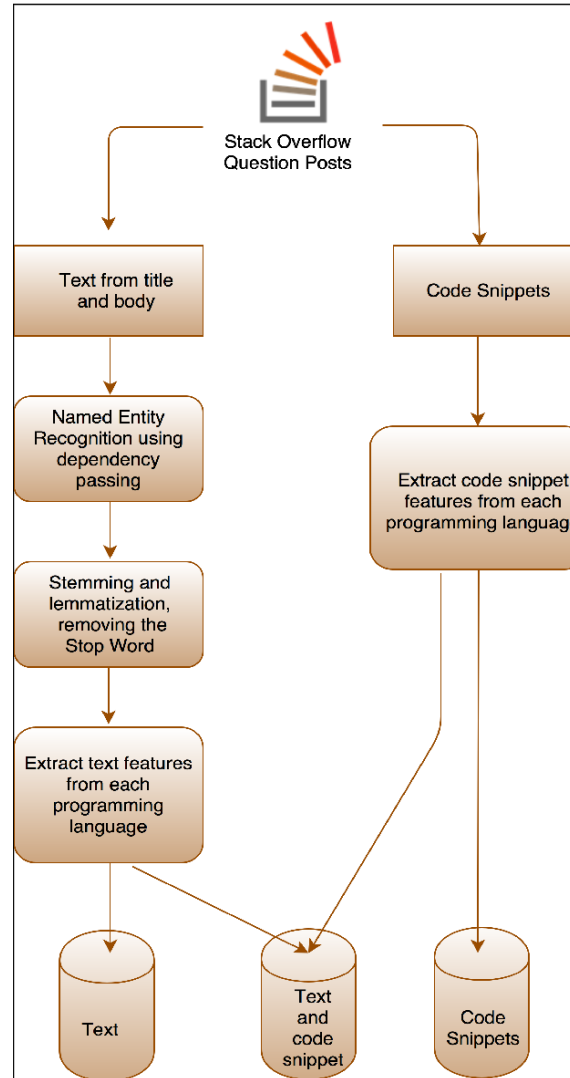
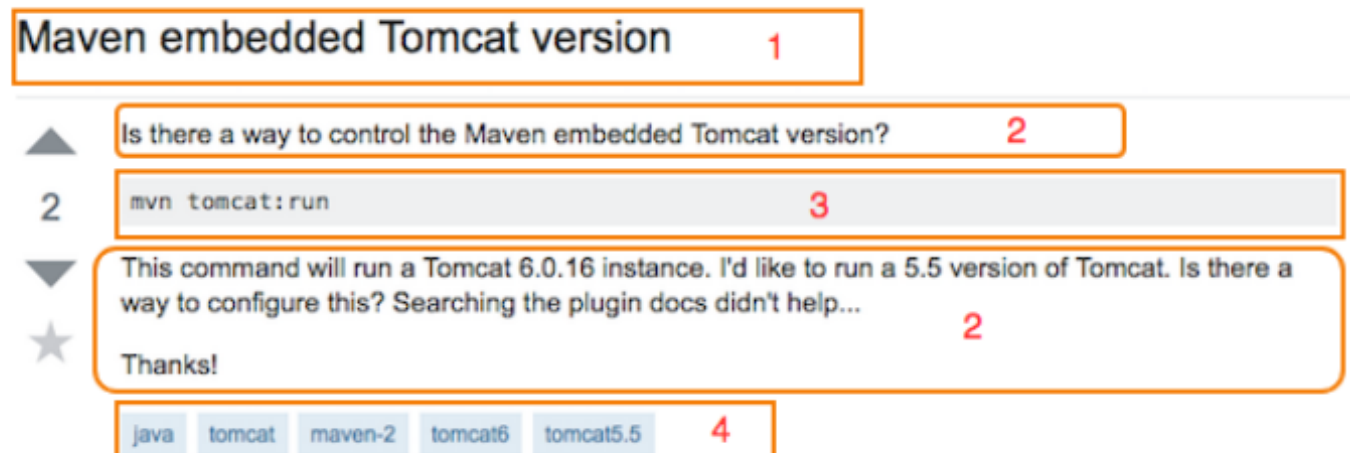


Fig. 2: The dataset extraction process

Dataset Extraction & Processing (continued)



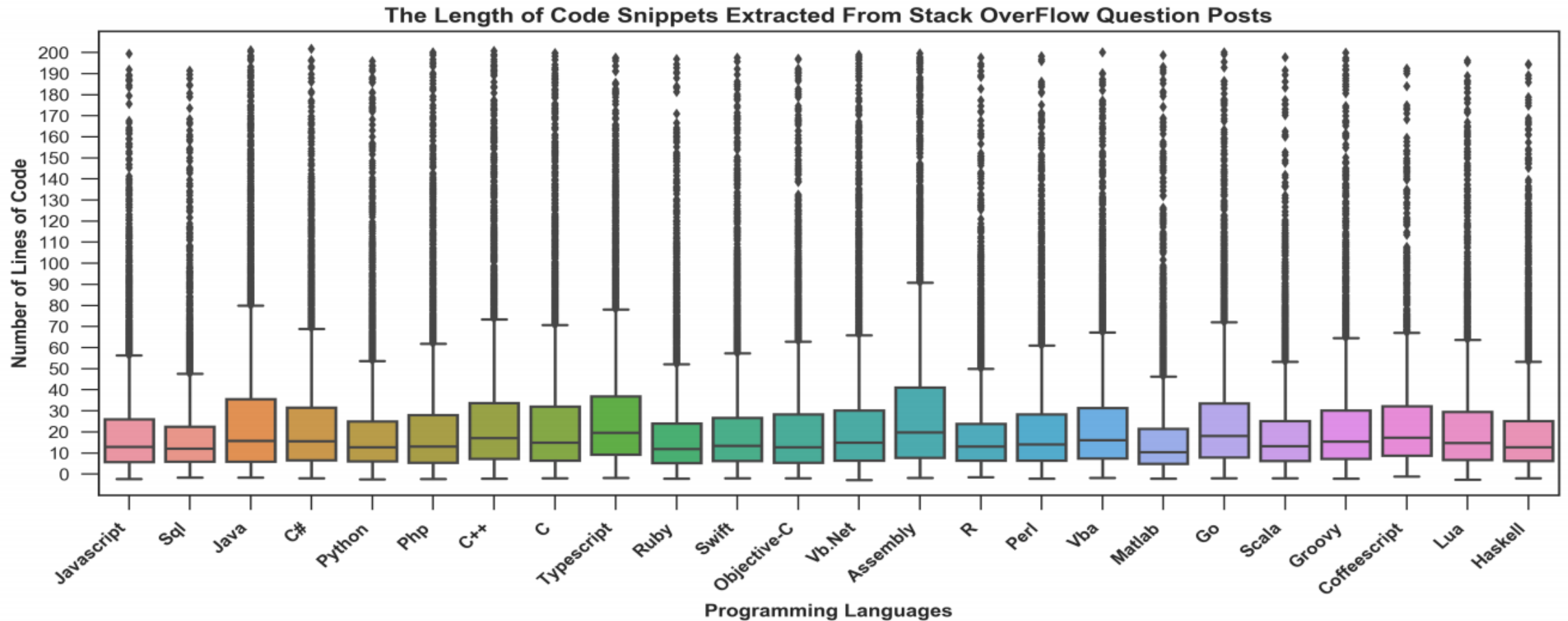
(a) Before applying NLP techniques.

maven tomcat version way maven tomcat version
command tomcat instance version tomcat way plugin
docs

(b) After applying NLP techniques.

Fig. 1: An example of a Stack Overflow Question.

Dataset Extraction & Processing (continued)



Methodology (Classifiers)

- Random Forest Classifier (RFC)
 - generates a number of decision trees from randomly selected subsets of training dataset
 - each subset provides a decision tree that votes to make the final decision
 - the final decision made depends on the decision of majority of trees (bagging)
 - a large number of trees in the forest give higher accuracy
 - if one or few of trees make a wrong decision, it will not affect the accuracy of the result significantly
 - avoids the overfitting problem seen in the Decision Tree model

Methodology (Classifiers)

- Extreme Gradient Boosting (XGBoost)
 - A tree-based model similar to Decision Tree and RFC
 - Modifies the weak learner to be a better learner
 - Each subtree makes the prediction sequentially
 - Each subtree learns from the mistakes that were made by the previous subtree
 - Parameters: minimum child weight, max depth, L1 and L2 regularization
 - Parameters were tuned using RandomSearchCV of Scikit-learn library.

Evaluation metrics

- Accuracy
 - = Correct predictions / Total predictions
 - = $(TP + TN) / (TP + TN + FP + FN)$
- Precision
 - True positive / Total Predicted positive
 - = $TP / (TP + FP)$
- Recall
 - True positive / Total Actual positive
 - = $TP / (TP + FN)$
- F1 Score = $2 \times \frac{Precision * Recall}{Precision + Recall}$

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Results

- **RQ1. Can we predict the programming language of a question in Stack Overflow?**

Classifier	Accuracy	Precision	Recall	F1 score
XGBoost	91.1%	0.91	0.91	0.91
RFC	86.3%	0.87	0.86	0.86

- Highest F1 score: Swift (0.97), GO (0.97), Groovy (0.97) and Coffeescript (0.97)
- Lowest F1 score: Java (0.75), SQL (0.78), C# (0.80) and Scala (0.88)

Results (continued)

- **RQ2. Can we predict the programming language of a question in Stack Overflow without using code snippets inside it?**

Classifier	Accuracy	Precision	Recall	F1 score
XGBoost	81.1%	0.83	0.81	0.81
RFC	75.6%	0.76	0.74	0.75

- 10% decrease in accuracy
- Highest F1 score: CoffeeScript (0.94), JavaScript (0.92), Swift (0.92), Go (0.92), Haskell (0.92), C (0.91) Objective-C (0.90) and Assembly (0.89)
- Lowest F1 score: Java, SQL (25% decrease in accuracy)

Results (continued)

- **RQ3. Can we predict the programming language of code snippets in Stack Overflow questions?**

Classifier	Accuracy	Precision	Recall	F1 score
XGBoost	77.7%	0.79	0.77	0.77
RFC	70.1%	0.72	0.72	0.70

- When the programming language of a code snippet is extremely hard to identity,
 - XGBoost frequently misclassified it as Objective-C,
 - RFC misclassified such snippets as Typescript.
- Highest F1 score: JavaScript (0.91), CoffeeScript (0.89) and PHP (.88)
- Objective-C has the worst F1 score and precision (0.56 and 0.42);

Results (continued)

Model	Description	Accuracy	Precision	Recall	F1 score
Previous					
Baquero [18] code snippets	A model trained using Support Vector Machine on question questions from Stack Overflow using code features	44.6%	0.45	0.44	0.44
Baquero [18] textual information	A model trained using Support Vector Machine on questions from Stack Overflow using text features	60.8%	0.68	0.60	0.60
Proposed					
code snippet features	XGBoost classifier trained on Stack Overflow questions using code snippet features	77.7%	0.79	0.77	0.78
textual information features	XGBoost classifier trained on Stack Overflow questions using textual information features	81.1%	0.83	0.81	0.81
code snippet and textual information features	XGBoost classifier trained on Stack Overflow questions using code snippets and textual information features	91.1%	0.91	0.91	0.91

Results (continued)

The Minimum Characters	Accuracy	Precision	Recall	F1-score
More than 10	77.7%	0.79	0.77	0.78
More than 25	79.1%	0.80	0.97	0.79
More than 50	81.7%	0.82	0.81	0.81
More than 75	83.1%	0.83	0.83	0.83
More than 100	84.7%	0.85	0.84	0.84

TABLE V: Effect of the minimum number of characters in code snippet on accuracy

Results (continued)

Programming	Precision	Recall	F1-score
Swift	0.98	0.96	0.97
Go	0.98	0.96	0.97
Groovy	0.99	0.95	0.97
Coffeescript	0.98	0.96	0.97
Javascript	0.97	0.95	0.96
C	0.98	0.95	0.96
C++	0.97	0.93	0.95
Objective-c	0.97	0.94	0.95
Assembly	0.96	0.95	0.95
Haskell	0.95	0.95	0.95
Python	0.97	0.91	0.94
Vb.net	0.95	0.93	0.94
PHP	0.94	0.91	0.93
Ruby	0.89	0.93	0.91
Perl	0.91	0.91	0.91
Matlab	0.92	0.90	0.91
R	0.91	0.89	0.90
Lua	0.94	0.86	0.90
Typescript	0.90	0.88	0.89
Vba	0.85	0.91	0.88
Scala	0.85	0.92	0.88
C#	0.81	0.79	0.80
CQL	0.73	0.85	0.78
Java	0.70	0.82	0.75

RQ1

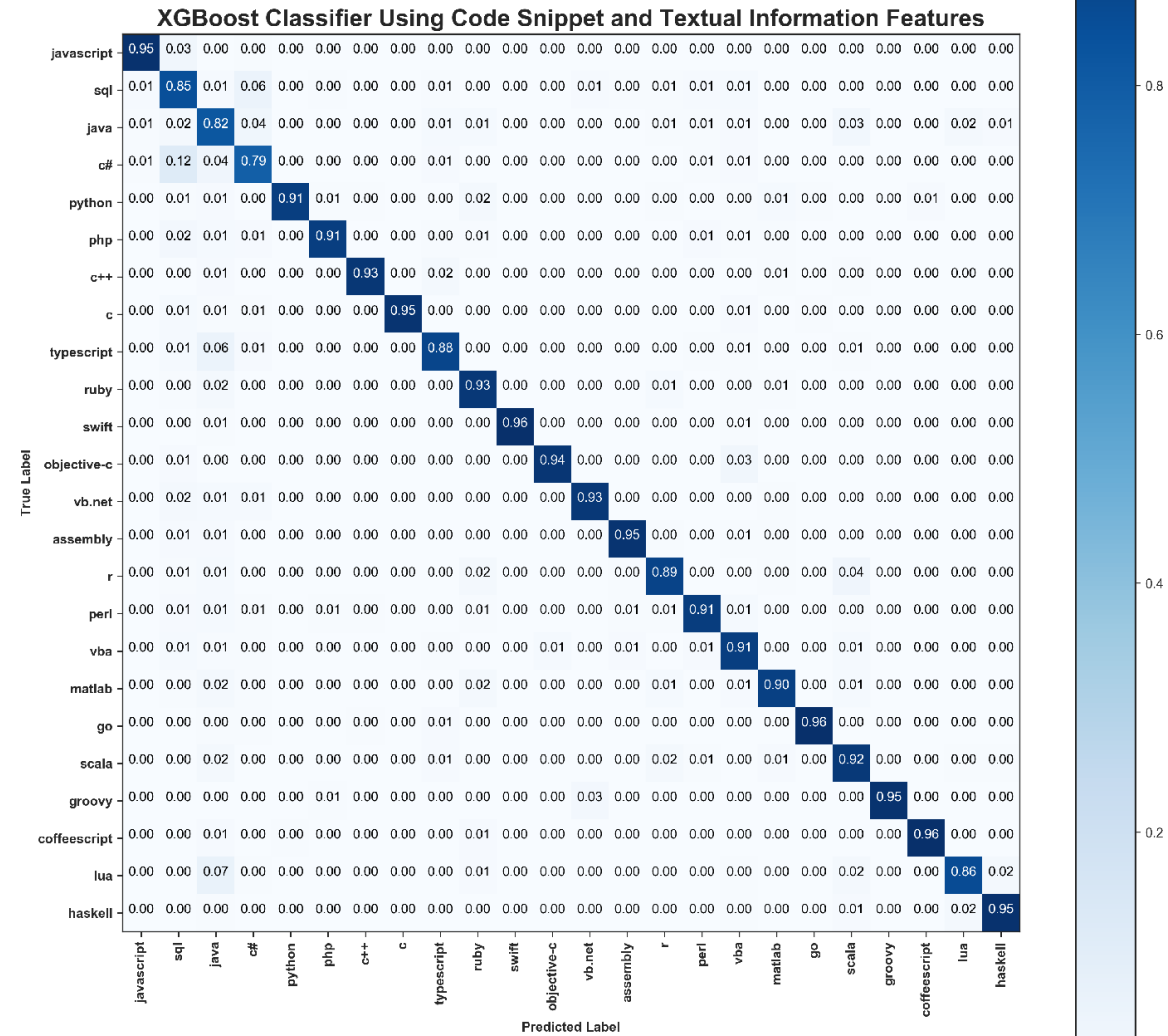
Programming	Precision	Recall	F1-score
Coffeescript	0.96	0.91	0.94
Javascript	0.94	0.89	0.92
Swift	0.94	0.89	0.92
Go	0.95	0.89	0.92
Haskell	0.92	0.91	0.92
C	0.93	0.88	0.91
Objective-c	0.94	0.87	0.90
Assembly	0.92	0.87	0.89
Python	0.95	0.82	0.88
Groovy	0.95	0.82	0.88
C++	0.92	0.83	0.87
Ruby	0.86	0.88	0.87
R	0.88	0.82	0.85
Perl	0.88	0.81	0.84
Matlab	0.88	0.80	0.84
Scala	0.80	0.90	0.84
Typescript	0.86	0.80	0.83
Vb.net	0.82	0.82	0.82
Vba	0.76	0.81	0.78
PHP	0.82	0.72	0.77
Lua	0.73	0.59	0.65
C#	0.65	0.61	0.63
SQL	0.42	0.75	0.54
Java	0.43	0.58	0.49

RQ2

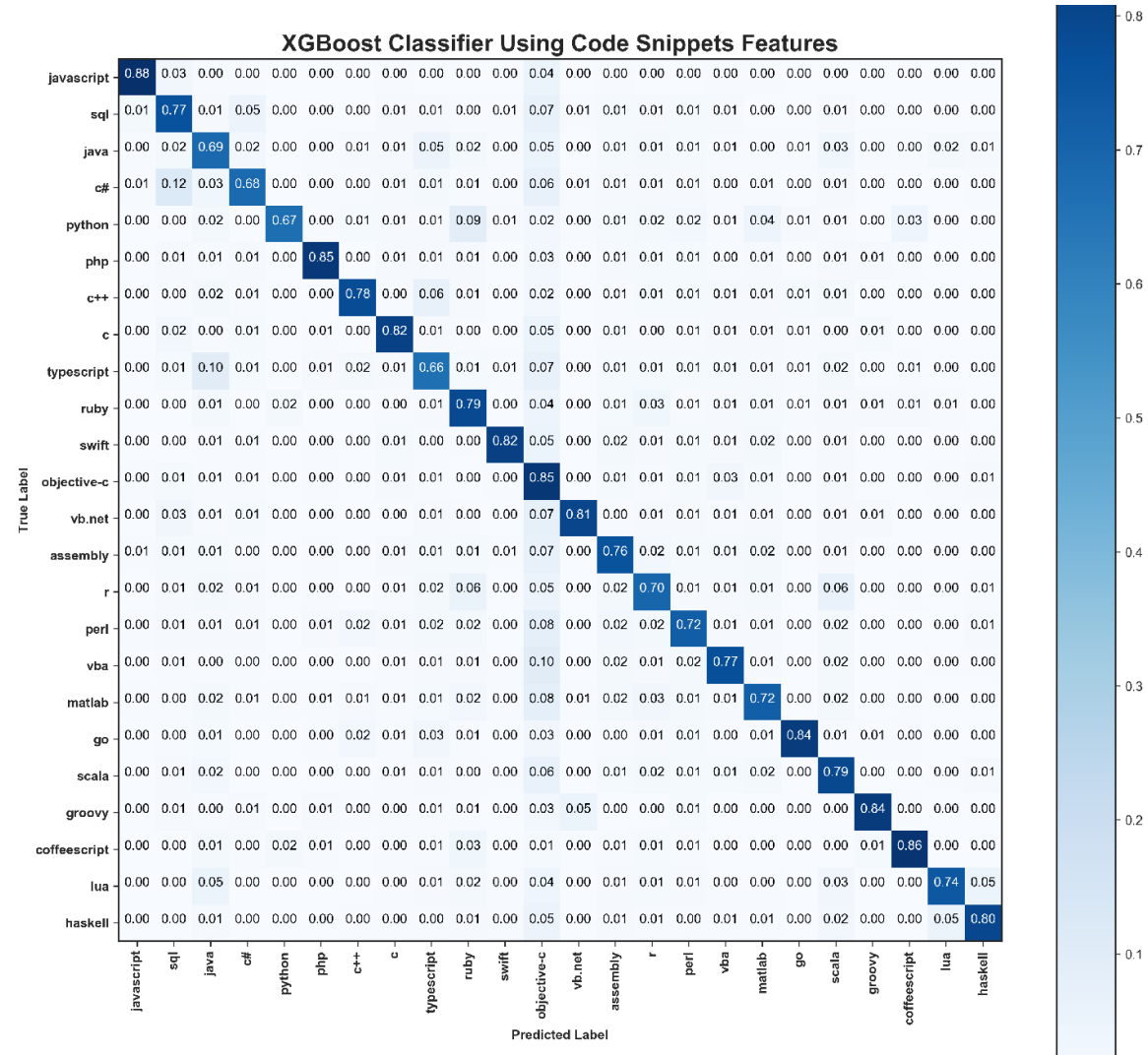
Programming	Precision	Recall	F1-score
Javascript	0.94	0.88	0.91
Coffeescript	0.92	0.86	0.89
PHP	0.91	0.85	0.88
Go	0.92	0.84	0.87
Groovy	0.91	0.84	0.87
Swift	0.91	0.82	0.86
C	0.86	0.82	0.84
Vb.net	0.89	0.81	0.84
Haskell	0.87	0.8	0.83
C++	0.86	0.78	0.82
Vba	0.82	0.77	0.80
Lua	0.87	0.74	0.80
Assembly	0.76	0.76	0.76
Python	0.85	0.67	0.75
Ruby	0.72	0.79	0.75
Matlab	0.79	0.72	0.75
Scala	0.71	0.79	0.75
SQL	0.70	0.77	0.73
C#	0.78	0.68	0.73
Perl	0.75	0.72	0.73
R	0.72	0.70	0.71
Typescript	0.68	0.66	0.67
Java	0.64	0.69	0.66
Objective-c	0.42	0.85	0.56

RQ3

Results (continued)



Results (continued)



Results (continued)

- Significant improvement in performance compared to previous approaches
 - Dependency parsing and extracting entity names using a Neural Network (NN) through Spacy
 - reduce noise
 - extract important features

Results (continued)

- Analysis of the feature space of the top performing languages
 - unique code snippet features (keywords/identifiers)
 - textual information features (libraries, functions)
 - Haskell: 'GHC', 'Ghci' (compilers), 'Yesod' (web-based framework) and 'Monad' (programming paradigm)
- Most top performing languages have a small feature space
- Java, Vba and C# have numerous libraries and standard functions having large feature space
- A large feature space adds more complexity to the ML models

Shortcomings

- Construct validity
 - Datasets were created based solely on programming language tags.
 - Language synonymous tags were left out.
 - For example, 'SQL SERVER', 'PLSQL' and 'MICROSOFT SQL SERVER' are related to 'SQL' but were discarded.
- Internal validity
 - Dependency parsing to extract entity may result in the loss of critical vocabulary.
 - Selecting additional features such as lines of code and programming paradigm could have improved results but was not considered.
- External validity
 - Stack Overflow was used in this study as the data source but other sources such as GitHub repositories were not explored.
 - Some common programming languages such as Cobol and Pascal were not considered in this study.

Conclusion & Future Work

- This work tackles the important problem of predicting programming languages from code snippets and textual information.
- NLP and ML techniques perform much better in predicting languages compared to tools that predict directly from code snippets.
- Results show that training and testing the classifier by combining the textual information and code snippet achieves higher accuracy rather than using either textual information or code snippets only.
- This classifier could be applied in scenarios such as code search engines and snippet management tools (Gists, Pastebin, QSnippets or Dash).
- In future, the authors are planning to apply CNN combined with Word2Vec to evaluate programming blog posts, library documentation and bug repositories.

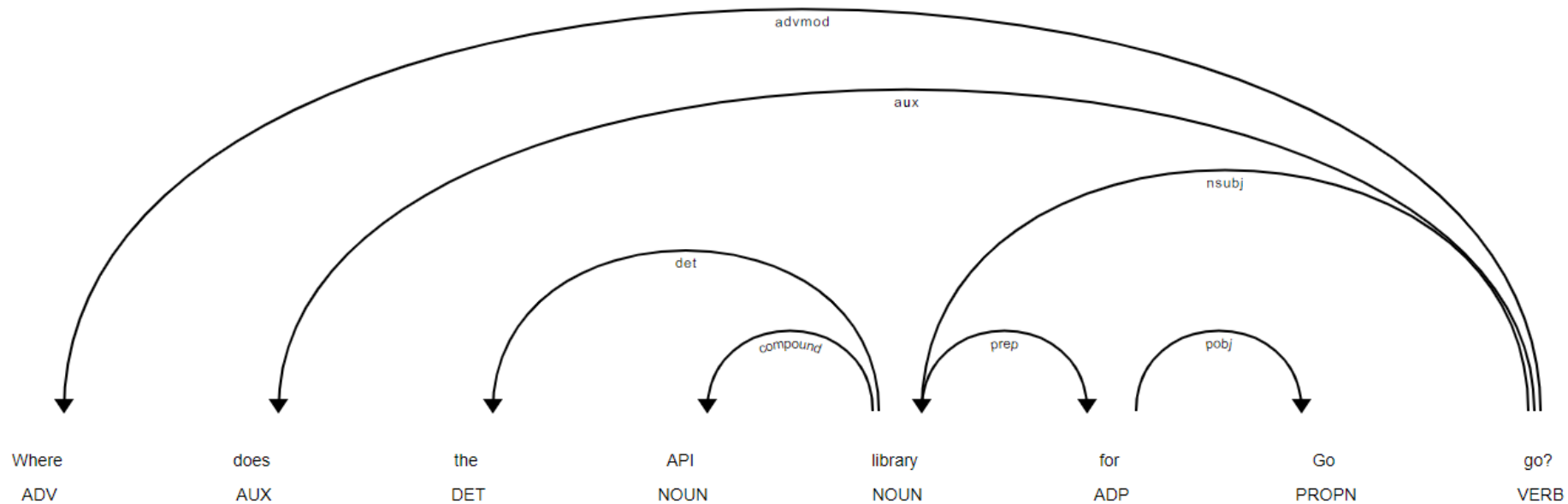
What I have learned reading this paper

- NLP Techniques:
 - Tokenization
 - Stemming & Lemmatization
 - Named entity Recognition
- ML Algorithms: RFC, XGBOOST
- Tools used:
 - XMLTODICT: .xml data parsing
 - BeautifulSoup: extract text and code snippets separately
 - Spacy: Named entity recognition
 - NLTK library: Stemming and Lemmatization
 - Scikit Learn library
 - RandomSearchCV: Parameter tune
 - Tf-IDF vectorizer: Dataset split

Spacy further explored

```
[11]: import spacy
      nlp = spacy.load("en_core_web_sm")

      from spacy import displacy
      doc = nlp("Where does the API library for Go go?")
      displacy.render(doc)
```



Spacy further explored

```
[19]: for t in nlp("Where to go to browse for open source go projects to work on?"):
      print(t, t.pos_, t.dep_)

      print("-----")

      for t in nlp("Where does the installed API library for Go go?"):
          print(t, t.pos_, t.dep_)
```

```
Where ADV advmod
to PART aux
go VERB csubj
to ADP aux
browse NOUN advcl
for ADP prep
open ADJ amod
source NOUN pobj
go VERB ROOT
projects NOUN dobj
to PART aux
work VERB relcl
on ADP prep
? PUNCT punct
-----
```

```
Where ADV advmod
does AUX aux
the DET det
installed VERB amod
API NOUN compound
library NOUN nsubj
for ADP prep
Go PROPN pobj
go VERB ROOT
? PUNCT punct
```

Spacy further explored

	SPACY	NLTK	ALLEN-NLP	STANFORD-NLP	TENSOR-FLOW
I'm a beginner and just getting started with NLP.	✓	✓	✗	✓	✗
I want to build an end-to-end production application.	✓	✗	✗	✗	✓
I want to try out different neural network architectures for NLP.	✗	✗	✓	✗	✓
I want to try the latest models with state-of-the-art accuracy.	✗	✗	✓	✓	✓
I want to train models from my own data.	✓	✓	✓	✓	✓
I want my application to be efficient on CPU.	✓	✓	✗	✗	✗