Transfer Learning in NLP

Gabriel Stella & Nicholas McKillip

Background

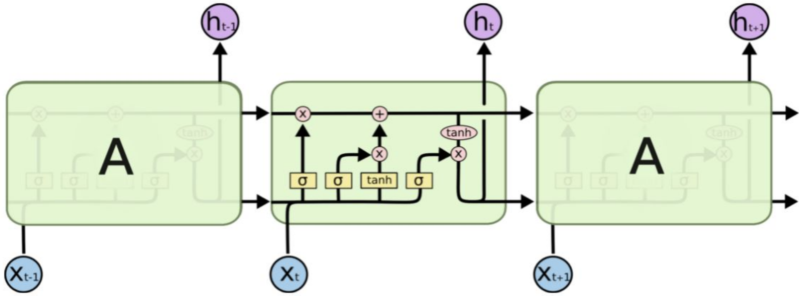
Training a general english language model→ Training for a specific task, starting from language model weights.

The most common approach in natural language processing has been only using linearly trained word vectors and initializing the actual model weights from scratch.

Motivations

- Deep models can be slow to train and sensitive to hyperparameters.
- By starting with a pre-trained language model, your model already understands the nature and semantics of text.
- This allows the model to converge much more quickly and achieve better end results

System



Train Deep LSTM network to be a language model on WikiText2

Begin with these same weights as the basis for a sentiment classification model on the IMDB dataset. We add a new linear output layer to predict sentiment.

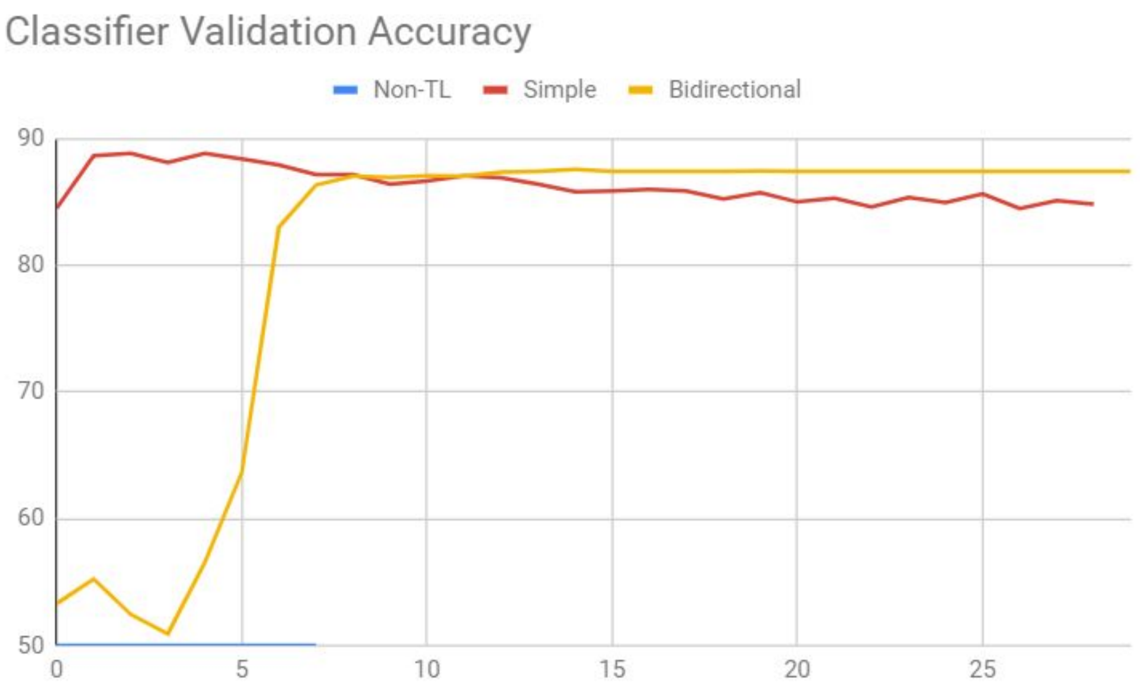
Iterations

1. Non-pretrained RNN model
2. Pretrained (transfer learning) RNN model
3. Multilayer/bidirectional RNN

- Dataset experiments

Results

Transfer learning completely surpassed the non-TL model



Conclusion

- Transfer learning was a success
- Slight problems with overfitting
- Bidirectional LSTM learned slowly but was more stable
- Non-TL model was essentially guessing (50% accuracy)
    - Specific dataset modifications allowed some learning

Sir Arthur Conan Doyle Probabilistic Language Model - with Part of Speech Tagging

By: Grant Weidner and Clayton Joseck

Project Briefing/Overview

- Goal: Determine if applying part of speech tags to the words of the corpus could improve the performance of a probabilistic language model.
- Used texts from Sir Arthur Conan Doyle’s *Sherlock Holmes* series
- Implements trigram probabilities and a POS tagger
- Outputs:
  - Given a two word input -> Predict the next word in the sequence
  - Generate a sentence given a random word in a document (machine learning)

Background and Motivation

- Considerations:
  - Doable yet interesting
  - Little background knowledge other a bit in lecture
- Motivations:
  - Work with machine learning
  - Develop something fun

Data Preprocessing

- Removed periods from titles (‘Mr’., ‘Mrs.’, ‘Dr.’, etc.) to avoid misinterpretation with end of sentence
- Separated each sentence into its own line to make trigram parsing easier
- End of sentence determined by a period, question mark, or exclamation point
- Cased-down all words to ensure they are interpreted as the same (e.g., Dog = dog = DOg)

Language Modeling

- Trigrams stored in dictionary of dictionaries
  - [(word1, word2)] [word3] = % of occurrences

- To retrieve most probable next word
  - Use key (word1, word2)
  - Loop over all word3’s
  - Choose word3 with highest probability of occurring

Part-of-Speech Tagging

- Chose to use NLTK POS tagger
  - Effective and reliable
  - Popular tagger
  - Strong documentation and examples
- To tag document:
  - Tokenize each word in document (NLTK tokenizer)
  - Tag each token (NLTK POS tagger)
  - Combine word with tag (e.g., book/NN) and write to tagged document file
- POS tagging increases overall accuracy of program

```
a/DT study/NN in/IN scarlet/JJ part/NN i/NN being/VBG a/DT reprint/NN from/
watson/JJ md/NN late/RB of/IN the/DT army/NN medical/JJ department/NN 2/CD
mr/NN sherlock/NN holmes/NNS in/IN the/DT year/NN 1878/CD i/NN took/VBD my/
having/VBG completed/VBN my/PRP$ studies/NNS there/EX i/NN was/VBD duly/RB
the/DT regiment/NN was/VBD stationed/VBN in/IN india/NN at/IN the/DT time/N
on/IN landing/NN at/IN bombay/NN i/NN learned/VBD that/IN my/PRP$ corps/NN
i/NN followed/VBD however/RB with/IN many/JJ other/JJ officers/NNS who/WP w
i/NN was/VBD removed/VBN from/IN my/PRP$ brigade/NN and/CC attached/VBN to/
there/EX i/NN was/VBD struck/VBN on/IN the/DT shoulder/NN by/IN a/DT jezail
i/NN should/MD have/VB fallen/VBN into/IN the/DT hands/NNS of/IN the/DT mur
here/RB i/JJ rallied/VBD and/CC had/VBD already/RB improved/VBN so/RB far/R
for/IN months/NNS my/PRP$ life/NN was/VBD despaired/VBN of/IN and/CC when/W
i/NN was/VBD dispatched/VBN accordingly/RB in/IN the/DT troopship/NN oronte
under/IN such/JJ circumstances/NNS i/VBP naturally/RB gravitated/VBN to/TO
there/EX i/VBZ stayed/VBD for/IN some/DT time/NN at/IN a/DT private/JJ hote
so/RB alarming/VBG did/VBD the/DT state/NN of/IN my/PRP$ finances/NNS becom
choosing/VBG the/DT latter/JJ alternative/JJ i/NN began/VBD by/IN making/VB
the/DT sight/NN of/IN a/DT friendly/JJ face/NN in/IN the/DT great/JJ wilder
in/IN old/JJ days/NNS stamford/NN had/VBD never/RB been/VBN a/DT particular
in/IN the/DT exuberance/NN of/IN my/PRP$ joy/NN i/NN asked/VBD him/PRP to/T
what/WP are/VBP you/PRP up/IN to/TO now/RB looking/VBG for/IN lodgings/NNS
trying/VBG to/TO solve/VB the/DT problem/NN as/IN to/TO whether/IN it/PRP i
i/NN should/MD prefer/VB having/VBG a/DT partner/NN to/TO being/VBG alone/R
you/PRP dont/VBP know/VB sherlock/NN holmes/NNS yet/RB he/PRP said/VBD perh
he/PRP is/VBZ a/DT little/JJ queer/NN in/IN his/PRP$ ideasan/JJ enthusiast/
as/RB far/RB as/IN i/NN know/VBP he/PRP is/VBZ a/DT decent/JJ fellow/NN eno
i/NN believe/VBP he/PRP is/VBZ well/RB up/RB in/IN anatomy/NN and/CC he/PRP
his/PRP$ studies/NNS are/VBP very/RB desultory/JJ and/CC eccentric/JJ but/C
if/IN i/JJ am/VBP to/TO lodge/VB with/IN anyone/NN i/NN should/MD prefer/VB
i/NN am/VBP not/RB strong/JJ enough/RB yet/RB to/TO stand/VB much/JJ noise/
i/NN had/VBD enough/NN of/IN both/DT in/IN afghanistan/NN to/TO last/JJ me/
how/WRB could/MD i/VB meet/VB this/DT friend/NN of/IN yours/NNS he/PRP is/V
he/PRP either/RB avoids/VBZ the/DT place/NN for/IN weeks/NNS or/CC else/RB
if/IN you/PRP like/IN we/PRP shall/MD drive/VB round/NN together/RB after/I
you/PRP proposed/VBD this/DT arrangement/NN so/IN you/PRP must/MD not/RB ho
it/PRP seems/VBZ to/TO me/PRP stamford/VB i/NN added/VBD looking/VBG hard/R
is/VBZ this/DT fellows/JJ temper/NN so/RB formidable/JJ or/CC what/WP is/VB
```

Results - Accuracy

- Tagged Model always performs better, by about 2%
- 100,000 trigrams per test iteration

Train/Test Split	Untagged Model	Tagged Model
95/5	62.33%	64.55%
90/10	57.03%	59.15%
85/15	54.74%	57.36%

Results - Sentence Generation - Good

- “good afternoon miss stoner said he this is a very fascinating and beautiful countryside”
- “it appears to hinge”
- “james mortimer the man whom we were all flecked and dashed with white his eyes were as unlike those of a precipice”
- “if they are all seaports”
- “i then proceeded down the half-rural villa-lined roads which lead to the front door”

Results - Sentence Generation - Bad

- “afterwards if i claim full justice in the wood and he in his disguise he packed them away in the air”
- “there is another man upon the left-hand side of the preceding days”
- “on the contrary she gave a cry of exultation or satisfaction upon his features dr”
- “you perceive that all was dark and the third demand during the day you understand”

Future Work

- Find optimal n-gram based on run-time and accuracy
- Explore more improvements to preprocessing
  - Helps resolve tokenizing and labeling errors
  - More capitalization and punctuation handling
- Optimize use of tags
  - Separate tag and word sequences
- Fun/Interesting Idea: Generate document and compare to Sir Arthur Conan Doyle
  - See how well our generation compares to the real literature
  - Could extend to more authors

Questions?

Forgery

A script writing bot
By Zachary Hughes and Michael Earl

Summary of Goals

- Idea: 1000 hours of Frazier bot
- Using IMSDB scripts, write its own script
  - ~2800 scripts
  - Mimic writers, hence name
  - Limited immediate use
    - Potential applications are infinite
      - Textbook writing

Input

Title management

- Need to be able to input title
- Given scripts
    - How do you get titles
        - Infinite possible setups
        - Quotes, what line it's on, etc
- Comparison magic
    - File extensions are useful
        - No spaces, but
    - Get substrings from script, compare

Input cont.

- Comparison magic
  - To get title, run string parser on first 50 lines of text
  - Compare each substring to file extension
    - Character by character
    - Batman 2 == batman2?
  - Recall with 50 lines: .706
    - 10 lines = .537
  - Unicode problems!!!
    - Names might have special characters, broke

Input cont. 2

Results:

- Store results in .txt
- Map function creates map from .txt for larger program
- Ex:
  - NAME#~ file.txt#~ [list of writers]#~

Problems:

- Weird titles, mismatched file extensions
  - The Dark Knight = batman2?

Input cont. 3

Problems cont.

- Without title, can’t get writers
- Use wrapper (TMDBsimple for The Movie Database v3 API) to get writers from title
    - Can’t always get writers from title even when have it
        - Likely problem with slight titling differences

Output

Markov chain

- Construct transitions of n-grams to following character
- Start generated text with random n-gram from the text
- Randomly (weighted based on occurrences) choose a transition of the last n-gram of generated text and add another character.
- If there is no such transition, add a random character.

Output cont.

Neural net

- Three layers of 500 LSTM nodes, each followed by a 20% dropoff to avoid overfitting.
- Feed n-gram transitions as input to the neural net.
- Train with batch size of 50 for 100 epochs over 20 hours.
- This approach gave bad results.

Examples

6-gram Markov on LOTR:

t down the Ferry..Pippin around the has endure. Then it, I mean not the ring for I much breath... CLOSE ON: RIVER lie still surprised. His treaches echo up from Gandalf's voice...Bill of the hours water...that once we have not. Merry and Saruman's voices...but through the blows the hilt. (CONT'D) I've ever you think. SAM No, Frodo is lost!! Frodo is longer looks at Strider drops the sounds his? How? SARUMAN Hunt thered riding smoke ring, as it on the Sea! Strong, the bottomless fear drive us? GANDALF (CONT'D) In the sound to his headed for two Hobbits a small sword slips on the drawn his staff... Bilbo's that...in the hobbits are hall. Go on, Gandalf steely light play across: she appled sunlit hillside and Sam watching from his spear...blacksmiths...a great 40 foot man-flesh. Saruman sits up in the one Dwarf blocks with tensils, provision at his gaze... BILBO (surprised) Not with down to tell me, Frodo, in a sad strength. He does not for far to the past the Party field. Bilbo instant

Examples cont.

12-gram Markov on LOTR:

hes! Frodo looks up as Aragorn towers over him. ARAGORN Frodo's face... GANDALF (V.O.) We must hold to his course west of the misty twilight world, past the foggy shapes of twisted trees. Somewhere behind him. ARWEN Why do you fear the past? You are Isildur's heir? LEGOLAS And heir to the throne of mountain kings. The world of Men. They're scattered everywhere. SAM What's the Elvish for friend? GANDALF All these long years we've been friends... slowly, faint lines appear like slender veins of luminous silver running through the air. MERRY (nervous) They're close. Frodo gasps in horror! The Shire is in ruins! The image suddenly clamp down on Merry and Pippin hurry through the trees near the fountain. They lie on soft couches as Elves leave food and wine for them. MOURNFUL SINGING drifts down from the passage, carrying Gandalf disappear into the steaming volcano. ELROND (V.O.) You have found your way to the last homely house east of the sea. The elves of Imladris have dwelt within this v

Examples cont.

4-gram Neural Net (trained for 20 hrs)

A
INGWRAITHS rider and shadow and

Interview Question Generation Application

CSCE 489-500

James "McLain" Johnson

Motivation

- Problem - Due to ease of access of applying to jobs online, companies can receive an overwhelming number of applicants. The hiring process needs to become more efficient in order to keep up with this volume.

- NLP offers many solutions to the hiring aspect of industry, but there is still much room for growth in the field. Particularly in areas previously thought to be too complicated to automate/enhance with NLP.

Abstract

I planned to implement an NLP application that would be able to accomplish two primary goals:

1. Accept several structures of input to allow use by both HR and potential employees.
2. Create logical and useful interview questions.

Original Solution: Encoder-Decoder neural net (Yuan et al., 2017)

Model I: POS tagger + pattern matching framework

Model II: POS tagger + web scraping

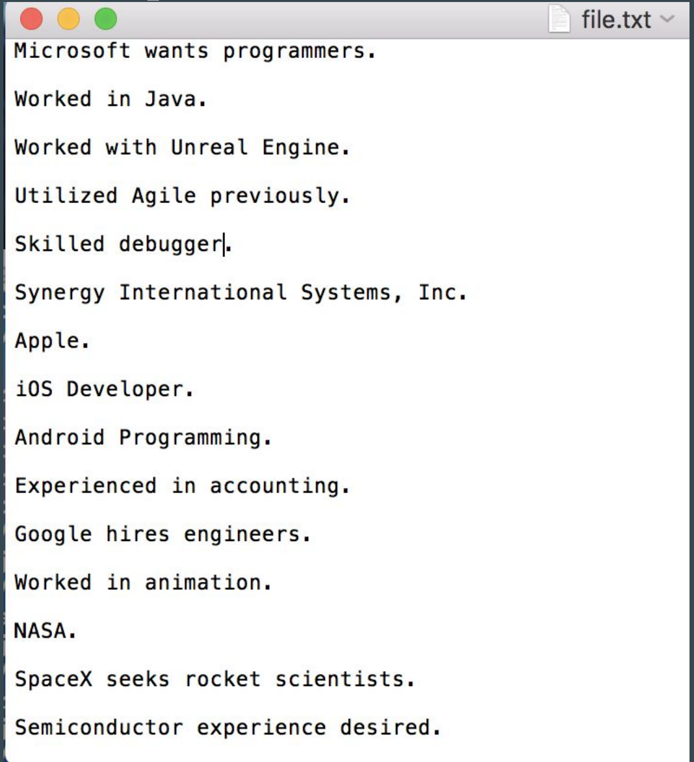
Original Design: Encoder-Decoder (Yuan et al., 2017)

- Cutting edge of the question generation field.
- Multi-Perspective Context Matching (Wang et al., 2016)

- However, required data in a form I did not have access to.
- Currently, the most interesting deep learning QG models are only able to generate fill-in-the-blank or simple restructuring of the sentence type questions.
- Does not result in “useful” questions very often.

Model I - Intro

- Take input file with either words of interest, or job description sentences.
- TextBlob used for POS tagger (~97% accuracy).
- Created POS patterns and question frames
- POS sequence of input pattern matched to produce question


file.txt

Microsoft wants programmers.

Worked in Java.

Worked with Unreal Engine.

Utilized Agile previously.

Skilled debugger.

Synergy International Systems, Inc.

Apple.

iOS Developer.

Android Programming.

Experienced in accounting.

Google hires engineers.

Worked in animation.

NASA.

SpaceX seeks rocket scientists.

Semiconductor experience desired.

Model I - Results

- Generated questions that were “logical” ~60% of the time, however hardly any could be considered “useful” interview questions.

  Ex: Input - “SpaceX seeks rocket scientists”

  Output - “What seeks SpaceX”

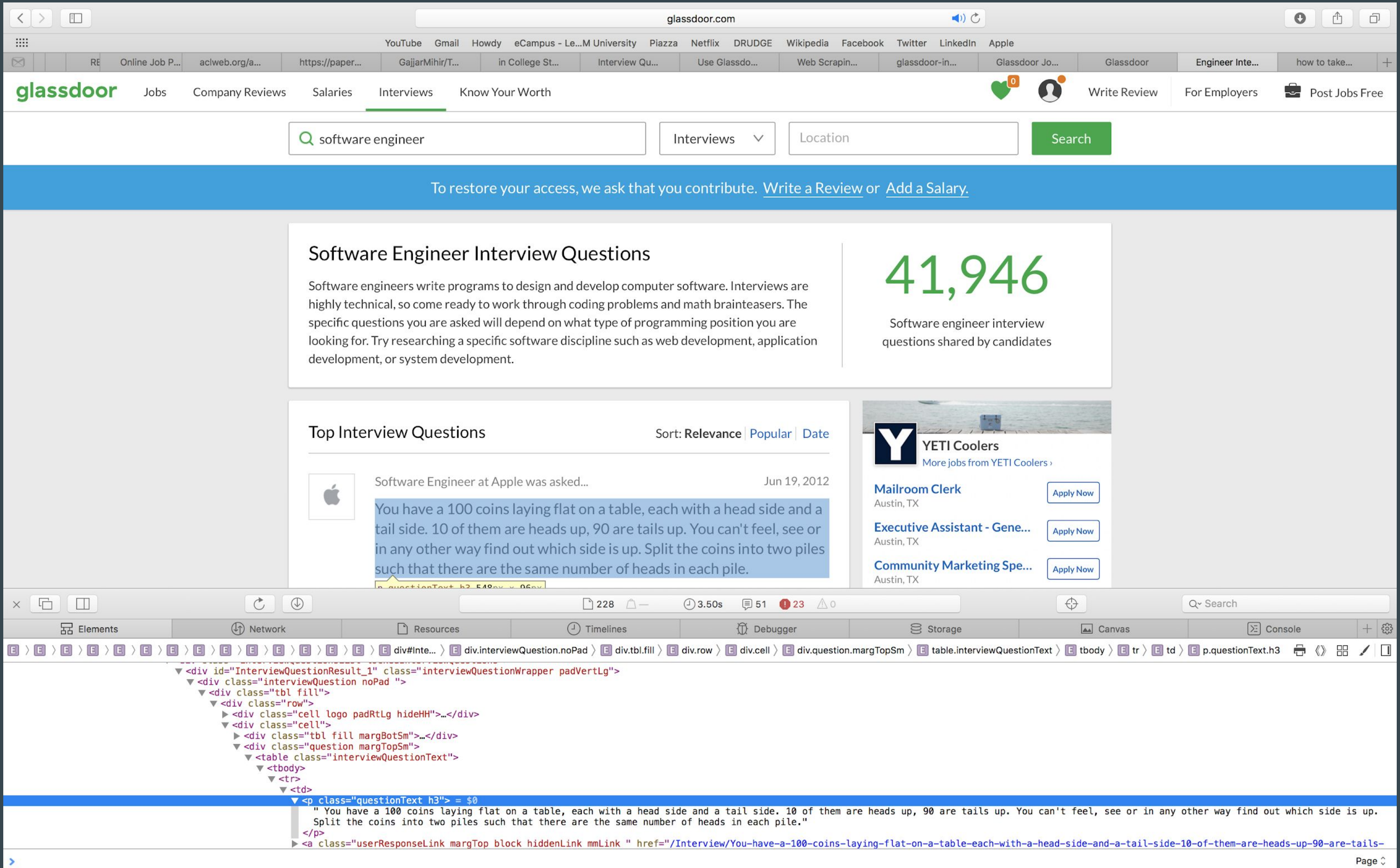
  Input - “NASA”

  Output - “What can you tell me about NASA?”

  Satisfied the first primary goal, but not the second.

Model II - Intro

- Kept the same data input method and TextBlob POS tagger from the first model.
- Created a program to take in a key phrase as input, and scrape user generated interview data from Glassdoor.com (used lxml)
- Combined the POS tagger from the first part to identify a key phrase/word from the input, and used it as the argument in the scraper.


glassdoor.com
YouTube Gmail Howdy eCampus - Le...M University Piazza Netflix DRUDGE Wikipedia Facebook Twitter LinkedIn Apple
glassdoor
Jobs
Company Reviews
Salaries
Interviews
Know Your Worth
Write Review
For Employers
Post Jobs Free
software engineer
Interviews
Location
Search
To restore your access, we ask that you contribute. Write a Review or Add a Salary.
Software Engineer Interview Questions
Software engineers write programs to design and develop computer software. Interviews are highly technical, so come ready to work through coding problems and math brainteasers. The specific questions you are asked will depend on what type of programming position you are looking for. Try researching a specific software discipline such as web development, application development, or system development.
41,946
Software engineer interview questions shared by candidates
Top Interview Questions
Sort: Relevance | Popular | Date
Software Engineer at Apple was asked...
Jun 19, 2012
You have a 100 coins laying flat on a table, each with a head side and a tail side. 10 of them are heads up, 90 are tails up. You can't feel, see or in any other way find out which side is up. Split the coins into two piles such that there are the same number of heads in each pile.
YETI Coolers
More jobs from YETI Coolers ›
Mailroom Clerk
Austin, TX
Apply Now
Executive Assistant - Gene...
Austin, TX
Apply Now
Community Marketing Spe...
Austin, TX
Apply Now
Elements
Network
Resources
Timelines
Debugger
Storage
Canvas
Console
div#Inte... div.interviewQuestion.noPad div.tbl.fill div.row div.cell div.question.margTopSm table.interviewQuestionText tbody tr td p.questionText.h3
<div id="InterviewQuestionResult_1" class="interviewQuestionWrapper padVertLg">
<div class="interviewQuestion noPad ">
<div class="tbl fill">
<div class="row">
<div class="cell logo padRtLg hideHH">…</div>
<div class="cell">
<div class="tbl fill margBotSm">…</div>
<div class="question margTopSm">
<table class="interviewQuestionText">
<tbody>
<tr>
<td>
<p class="questionText h3"> = $0
" You have a 100 coins laying flat on a table, each with a head side and a tail side. 10 of them are heads up, 90 are tails up. You can't feel, see or in any other way find out which side is up. Split the coins into two piles such that there are the same number of heads in each pile."
</p>
<a class="userResponseLink margTop block hiddenLink mmLink " href="/Interview/You-have-a-100-coins-laying-flat-on-a-table-each-with-a-head-side-and-a-tail-side-10-of-them-are-heads-up-90-are-tails-

Model II - Results

- The outputted questions had a much higher rate of being both logical, and useful.
- Several interview questions returned per request.

Ex: Input - “SpaceX seeks rocket scientists”

Output - “What are the formulas for specific impulse, characteristic velocity, and thrust?”

Input - “NASA”

Output - “What is the function of integrated circuit?”

Satisfied both primary goals.

Conclusion

- The second model performed much better than the first model for this application.
- Question Generation vs Question Procurement
- Deep learning avenues are still worth exploring given enough time to gather the appropriate data.
- For the meantime, the combination of some NLP tasks such as POS tagging, and human generated questions seems to be the strongest type of model, and accomplishes the primary goals for this application the best.

Extractive Text Summarization

BY: ZACK CHRISTIE

Approach to Summarizing Article From the Web

- **Extract article text from html page**
- **Tokenize each sentence from the article**
- **Tokenize each word in each sentence**
- **Filter out punctuation from tokenized sentences**
- **Filter out stop words from tokenized sentences**
- **Create frequency table from the tokenized words**
- **Pass each word through a stemmer while creating frequency table**
- **Calculate each sentence weight**
- **Build summary**

Extract article text from html page

- Used a library called beautiful soup and lxml.
- Beautiful soup would take in any valid URL and make it into an lxml format
  - Once in lxml the p tags could be extracted for use in the rest of the problem.

Tokenize each sentence from the article

In order to extract whole sentences from the article in the original form, tokenizing the sentences was necessary.

- Another reason for tokenizing the sentences is that it will increase the accuracy of the word tokenizing by excluding some of the punctuation.

Example: This is one sentence. This is a second sentence. This is a third sentence

Tokenized: [This is one sentence., this is a second sentence., this is a third sentence.]

Tokenize each word in each sentence

Iterate through all the tokenized sentences and split the sentences by tokenizing each word within that sentence.

After tokenizing the words I will filter out all the punctuation instances

After filtering out all the punctuation instances I also filter out all stop words.

Example:

Sentence: Houston is the most populous city in Texas and the fourth largest in the U.S., while San Antonio is the second-most populous in the state and seventh largest in the U.S.

Tokenized Sentence: ['Houston', 'is', 'the', 'most', 'populous', 'city', 'in', 'Texas', 'and', 'the', 'fourth', 'largest', 'in', 'the', 'U.S.', ',', 'while', 'San', 'Antonio', 'is', 'the', 'second-most', 'populous', 'in', 'the', 'state', 'and', 'seventh', 'largest', 'in', 'the', 'U.S', '.']

Stop Words Removed: houston populous city texas fourth largest u.s. , san antonio second-most populous state seventh largest u.s .

Stop Words Removed and punctuation removed: houston populous city texas fourth largest san antonio populous state seventh largest

Something worth noting here in removing the punctuation we also remove the words “second-most” and “u.s”

Create frequency table from the tokenized words

After the tokenizing and filtering, the words are then passed through a stemmer and used to create a frequency table.

Using previous example: houston populous city texas fourth largest san antonio populous state seventh largest.

The table on the left is the word counts. The table on the right is the word count divided by the largest count to come up with the frequency.

Word	Frequency
'houston'	1
'populous'	2
'city'	1
'texas'	1
'fourth'	1
'largest'	2
'san'	1
'antonio'	1
'state'	1
'seven	1

Word	Frequency
'houston'	0.5
'populous'	1
'city'	0.5
'texas'	0.5
'fourth'	0.5
'largest'	1
'san'	0.5
'antonio'	0.5
'state'	0.5
'seven	0.5

Calculate each sentence weight

To calculate the sentence weight, I iterate through all the words in the sentence if the word is in the frequency table and make the minimum and maximum weight cut that words weight will be added to the sentence weight.

Example using previous sentence: <u>Houston</u> is the most <u>populous</u> <u>city</u> in <u>Texas</u> and the <u>fourth</u> <u>largest</u> in the U.S., while <u>San</u> <u>Antonio</u> is the second-most populous in the <u>state</u> and <u>seventh</u> largest in the U.S.

Sentence Weight = 0.5 + 1 + 0.5 + 0.5 + 0.5 + 1 + 0.5 + 0.5 + 0.5 + 0.5 = 6

Minimum requirements for word weight in my implementation is 0.1 and they cannot exceed 0.9. This is so I do not allow non relevant words to effect the sentence weight.

Build Summary

After all the sentence weights are calculated, I the construct the summary.

This can happen two different ways, if the command line has a number greater than 0 or if its 0 itself.

If the command line parameter is 7, then the summary will have the 7 highest weight sentences in it.

If the command line parameter is 0, then only the sentences with a weight of 3x the sentence weight average will be included.

- Example sentence weight average: 1, sentence one weight: 1, sentence two weight: 2, sentence three weight: 3
- So only sentence three will be in the summary

Not so good precision on test case

A South Carolina police chief who lost her job two years ago because she is gay has been fired again.

Latta Town Administrator Jarrett Taylor, who helped reinstate Crystal Moore's job in 2014, said Thursday's firing had nothing to do with her sexual orientation.

"It's aggravating," Taylor said Friday. "I defended her so strongly a couple of years ago. I still stand by my actions then. But things have changed."

Moore made some poor decisions and there were administrative tasks that weren't done on time, according to Taylor.

Man with machete chases after clown spotted near woods in NC

She was suspended for five days last month after a series of mistakes, including failing to inform supervisors of a sexual harassment claim between two other employees as soon as it was reported. She also revealed an officer's salary at a public meeting and checked another employee's disciplinary records without permission.

Moore posted on her Facebook page that she was shocked and angered by the dismissal. Recently diagnosed with a cancerous tumor on her thumb, she is also is running for sheriff in Dillon County.

"This won't stop me from serving the people of Latta or Dillon County," she wrote. "This is a tough time for all of us, but I know justice will prevail."

"I'm seriously sorry this came at an inopportune time in her life," Taylor said. "But at some point, the job has to be done right."

Eleven-year-old believed to have died playing 'Choking Game'

Taylor helped Moore two years ago when Latta Mayor Earl Bullard issued seven reprimands and fired her. Taylor thought the disciplinary actions were bogus and released a tape he made of a conversation with the mayor.

"I'd much rather have somebody who drank and drank too much taking care of my child than I had somebody whose lifestyle is questionable around children, because that ain't the damn way it's supposed to be," Bullard said on tape.

The town voted to strip the mayor of his power and the Town Council hired Moore back. Taylor said he wanted Moore to do well and is still glad he helped her out.

"This has nothing to do with what she does in her life," he said. "I don't care as long as things get done right. This has been an ongoing issue. We asked her to change and gave her a chance to change and she just didn't."

My Summary:

Latta Town Administrator Jarrett Taylor, who helped reinstate Crystal Moore's job in 2014, said Thursday's firing had nothing to do with her sexual orientation.

Given Summary:

A South Carolina police chief who lost her job two years ago because she was gay has been fired again.

Analysis of Summaries from Previous Slide

My Summary:

Latta Town Administrator Jarrett Taylor, who helped reinstate Crystal Moore's job in 2014, said Thursday's firing had nothing to do with her sexual orientation.

Given Summary:

A South Carolina police chief who lost her job two years ago because she was gay has been fired again.

Rouge-1: {'f': 0.13636363140495886, 'p': 0.125, 'r': 0.15}

Rouge-2: {'f': 0.0, 'p': 0.0, 'r': 0.0}

Rouge-l: {'f': 0.08944281524862291, 'p’: 0.08333333333333333, 'r': 0.1}

Good precision on test case

The U.S. Coast Guard searched Friday for two Marine helicopters that collided with 12 people on board near the Hawaiian island of Oahu.

Search conditions were challenging because of darkness and high surf, Coast Guard Chief Petty Officer Sara Mooers told Los Angeles radio station KNX-AM. She said a high surf advisory was in effect for waves 10 to 15 feet building throughout the morning.

Coast Guard District 14 told CBS News the debris field included an empty life raft and fire on the water.

The transport helicopters each had a crew of six from Marine Corps Base Hawaii and crashed just before midnight Thursday, officials said. No other passengers were aboard the CH-53E "Super Stallions," which came from the 1st Marine Aircraft Wing, Marine Capt. Timothy Irish said.

The aircraft were taking part in a nighttime training mission. It's unclear what caused the crash.

A Coast Guard helicopter and C-130 airplane spotted the debris field 2 1/2 miles offshore early Friday. The debris covers an area of 2 miles, Irish said.

The search includes aircraft from the Navy and Air Force, a Honolulu Fire Department rescue boat and Coast Guard cutters, officials said.

"It is a true search-and-rescue effort, and it is ongoing," Irish said just before daybreak on Oahu, where a steady rain was falling on the North Shore.

The collision comes less than a year after the Marine Corps' new hybridized airplane-and-helicopter aircraft crashed during a training exercise, killing two Marines. The MV-22 Osprey went down last May with 21 Marines and a Navy corpsman on board. In 2011, one serviceman was killed and three others were injured when a CH-53D Sea Stallion chopper crashed in Kaneohe Bay, Hawaii.

Fox News' Jennifer Griffin and The Associated Press contributed to this report.

My Summary:

The U.S. Coast Guard searched Friday for two Marine helicopters that collided with 12 people on board near the Hawaiian island of Oahu.

Given Summary:

The U.S. Coast Guard searched Friday for two Marine helicopters that collided with 12 people on board near the Hawaiian island of Oahu.

Analysis of Summaries from Previous Slide

My Summary:

The U.S. Coast Guard searched Friday for two Marine helicopters that collided with 12 people on board near the Hawaiian island of Oahu.

Given Summary:

The U.S. Coast Guard searched Friday for two Marine helicopters that collided with 12 people on board near the Hawaiian island of Oahu.

.

Rouge-1: {'f': 0.999999995, 'p': 1.0, 'r': 1.0}

Rouge-2: {'f': 0.999999995, 'p': 1.0, 'r': 1.0}

Rouge-l: {'f': 0.9999999999995, 'p': 1.0, 'r': 1.0}

Overall average of ROGUE-N testing for over 400 articles

Articles are provided by CORNELL NEWSROOM

Rouge-1 avg: f: 0.43256197711874755 p: 0.42072589748135314 r: 0.5528223741985308

Rouge-2 avg: f: 0.3442072806492886 p: 0.34535343815801556 r: 0.44189773024408724

Rouge-l avg: f: 0.3784543279555752 p: 0.4087441079678198 r: 0.5334969981566555