

Определение наилучшего ответа на StackOverflow

Никита Подгузов

Научный руководитель: Рауф Курбанов

Санкт-Петербургский Академический университет

30 марта 2018 года

Возможности сервисов вопросов и ответов:

- Задавать вопрос (и отмечать правильный ответ)
- Отвечать на вопросы, заданные другими пользователями
- Голосовать за понравившиеся ответы

YAHOO! ANSWERS

Quora

 stackoverflow

Особенности системы:

- Узкоспециализированная
- Большая база вопросов
- Наличие сниппетов кода в вопросах и ответах

The screenshot shows a Stack Overflow page for the question "What is the \"-->\" operator in C++?". The page includes a search bar, navigation links (Questions, Developer Jobs, Tags, Users), and a "Sign Up" button. The question text states that the user was surprised by a code snippet that compiled in Visual Studio 2008 and GCC 4.4. The code snippet is as follows:

```
#include <stdio.h>
int main()
{
    int x = 10;
    while (x--> 0) // x goes to 0
    {
        printf("Id ", x);
    }
}
```

The question is marked as "asked 8 years, 5 months ago", "viewed 595,100 times", and "active 20 days ago". It has a "BLOG" section with a link to "Quantum Computing Site Launches with the Help of Strangeworks".

The question has 21 answers. The top answer, with a score of 7204, is marked as the best answer with a green checkmark. It explains that the conditional code decrements `x`, while returning `x`'s original (not decremented) value, and then compares the original value with `0` using the `>` operator. The answer also provides a better understanding of the statement, showing it can be written as `while((x--) > 0)`.

The second answer, with a score of 2183, explains that the operator is a very complicated operator, so even ISO/IEC JTC1 (Joint Technical Committee 1) placed its description in two different parts of the C++ Standard. It also mentions that the operator is described in §5.2.6/2 and §5.9 of the C++03 Standard.

Введение

Постановка задачи

Проблемы:

- Большая доля "неразрешенных" вопросов
- Нет возможности помочь оценить правильность ответов пользователю, задавшему новый вопрос

Хотим научиться определять правильные ответы, используя базу вопросов StackOverflow

Введение

Google & StackOverflow

Google ruby capitalize

About 369.000 results (0,34 seconds)

Capitalize first letter in ruby - Stack Overflow
<https://stackoverflow.com/questions/3724913/capitalize-first-letter-in-ruby>
Sep 16, 2010 - Unfortunately, it is impossible for a machine to upcase/downcase/capitalize properly. ... That's why Ruby's String class only supports capitalization for ASCII characters, because there it's at least somewhat well-defined.

How to convert a string to lower or upper case in Ruby	21 Dec 2014
ruby - Using 'capitalize' or 'capitalize!'	29 Sep 2014
Capitalize the first letter of each word - Ruby	26 Jan 2014
Capitalize only first character of string and leave others alone ...	12 Nov 2011

More results from stackoverflow.com

Class: String (Ruby 2.2.0) - Ruby-Doc.org
<https://ruby-doc.org/core-2.2.0/String.html>
Jump to **capitalize** - capitalize => new_str click to toggle source. Returns a copy of str with the first character converted to uppercase and the remainder to lowercase. Note: case conversion is effective only in ASCII region. "hello".capitalize #=> "Hello" "HELLO".capitalize #=> "Hello" "123ABC".capitalize #=> "123abc".

```
::try_convert - capitalize! - chomp - gsub
```

Old version

Google ruby capitalize

Alle Nieuws Shopping Afbeeldingen Video's Meer Instellingen Tools

Ongeveer 370.000 resultaten (0,34 seconden)

Capitalize first letter in ruby - Stack Overflow
<https://stackoverflow.com/questions/3724913/capitalize-first-letter-in-ruby>
The upcase method capitalizes the entire string. I need to capitalize only the first letter. Also, I need to support several popular languages, like German and Russian. How do I do it?

7 antwoorden

Best Answer 197 votes	Antwoord 2 van 7 37 votes	Antwoord 3 van 7 18 votes
It depends on Ruby version you use. Ruby 2.4 and higher it just works, as since this version ruby supports Unicode case mapping. "wapw".capitalize	capitalize first letter of first word of string "kirk douglas".capitalize #=> "Kirk douglas" capitalize first letter of each word in rails: "kirk"	Unfortunately, it is impossible for a machine to upcase/downcase/capitalize properly. It needs way contextual information

Class: String (Ruby 2.2.0) - Ruby-Doc.org
<https://ruby-doc.org/core-2.2.0/String.html> - Vertaal deze pagina
Spring naar **capitalize** - capitalize => new_str click to toggle source. Returns a copy of str with the first character converted to uppercase and the remainder to lowercase. Note: case conversion is effective only in ASCII region. "hello".capitalize #=> "Hello" "HELLO".capitalize #=> "Hello" "123ABC".capitalize ...

```
::try_convert - #[] - count - dump
```

New version

Введение

Обзор имеющихся решений

"Towards Predicting the Best Answers in CB QAS" (Tian et al. 2013)

- Три вида фичей: $A \leftrightarrow A$, $A \leftrightarrow Q$, A
- Использование Vector Space Model + TF-IDF для определения похожести
- Использование лингвистических фичей (длина текста, количество предложений, читаемость и др.)
- Учитывается лишь наличие/отсутствие сниппетов кода
- Random Forest Classifier

Введение

Обзор имеющихся решений

"State of the art Best Answer Prediction based on Discretisation of Shallow Linguistic Features" (Gkotsis et al. 2014),

"Moving to Stack Overflow: Best-Answer Prediction in Legacy Developer Forums" (Calefato et al. 2016)

- Четыре вида фичей: $A \leftrightarrow A$, A , *user-rating* и *answer-rating*, *thread*
- Использование лингвистических фичей (длина текста, количество предложений, читаемость и др.)
- Использование вероятностной униграммной модели для оценки вероятности ответа
- Использование группировки ответов и дискретизации фичей
- Не учитывает сниппеты кода
- Alternating Decision Tree Classifier

Введение

Минусы имеющихся решений

- Не используется текст вопроса
- Не учитывается порядок слов в предложении
- Не учитываются синонимы и похожие слова, то есть игнорируется семантика
- Не используется содержание сниппетов кода

Цели и задачи

Цель: научиться определять правильность ответа на StackOverflow, используя как его текст, так и код, который может присутствовать внутри ответа

Задачи:

- Реализовать классификатор на основе нейронных сетей, использующий текст ответов
- Добавить использование сниппетов кода в классификаторе
- Сравнить результаты с имеющимися работами
- Проанализировать влияние наличия фичей от сниппетов кода на точность классификации

Данные

Общие факты

Данные:

- Дамп базы вопросов StackOverflow
- XML-файл размером $\sim 50GB$
- Новые данные могут быть получены с помощью API
- Формат файла: *type_id, id, score, date, body*

Данные

Анализ и обработка

Анализ:

- 40 миллионов постов, из них 16 миллионов вопросов и 24 миллионов ответов
- 7 миллионов вопросов (47%) без отмеченного правильного ответа
- 2 миллиона вопросов (13%), у которых нет ни одного ответа
- 21.5 миллионов постов (54%), в которых присутствуют сниппеты кода.

Обработка:

- Удаляем вопросы с рейтингом ≤ 0 , а также вопросы, у которых нет ни одного ответа
- Из остальных постов извлекаем его *body*
- Сохраняем весь код, находящийся в тегах `<code>`
- Очищаем от тегов и сохраняем весь остальной текст вопроса/ответа

После обработки получили 6.5 миллионов вопросов, из них 2 миллиона вопросов (31%) без правильного ответа, а также 13.5 миллионов ответов

Подходы к анализу текста

Bag of words

Идея:

- Каждому слову сопоставляем вектор длины, равной размеру словаря
- Документ: сумма векторов слов

Проблемы:

- Не учитывается семантика
- Не учитывается порядок слов
- Большая размерность

Подходы к анализу текста

Bag of words (пример)

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

Stopword List

for
is
of
the
to

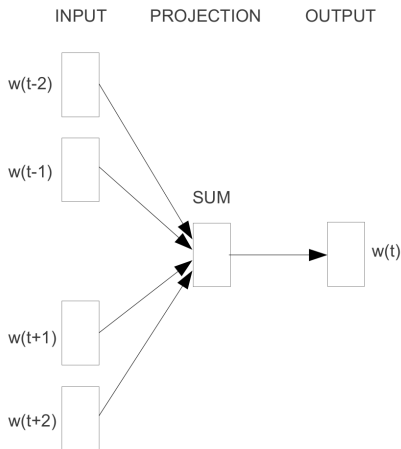
Подходы к анализу текста

Word2Vec

Идея:

- Каждому слову сопоставляем вектор фиксированной длины
- Обучаем на неразмеченном корпусе текстов CBOW/Skip-gram архитектуру
- Вектор отражает смысл слова, сохраняется семантика

Чтобы учесть специфику технического языка, обучаться лучше на текстах `StackOverflow`



CBOW

Подходы к анализу текста

Рекуррентные нейронные сети

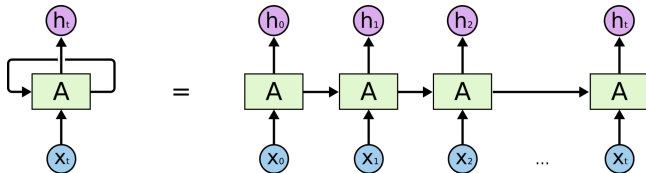
Хотим научиться учитывать порядок слов в тексте

Идея:

- Используем embedding слов из Word2Vec
- Отдаем на вход клетке сети новое слово и выход с предыдущей (учет контекста)
- Хотим, чтобы выход сети отражал смысл входного текста

Одна из двух основных архитектур: *LSTM*-клетка

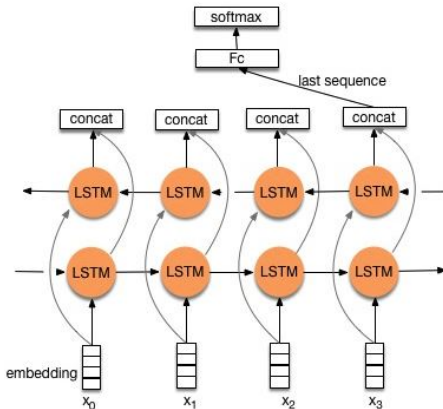
Также используются двунаправленные рекуррентные нейронные сети для захвата контекста справа



Классификация текстов

Базовая архитектура

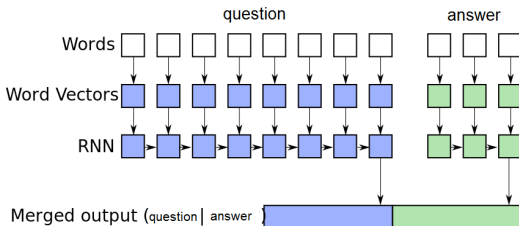
- Классификация ответов на два класса: правильный/неправильный
- В качестве embedding-а используется векторное представление, обученное на текстах вопросов и ответов StackOverflow



Классификация ответов

Учет текста вопроса

- Хотим также учитывать текст вопроса, чтобы понимать релевантность ответа
- Добавим *BiLSTM*-сеть для текста вопроса и будем использовать эти признаки вместе с признаками ответа



Подходы к анализу кода

Проблемы

Код похож на текст, поэтому можно попробовать применить аналогичные методы

Проблема: нелинейная структура кода (циклы, ветвления и др.)

Замечание: тем не менее, может хорошо сработать для однострочных сниппетов

Подходы к анализу кода

Использование синтаксического дерева

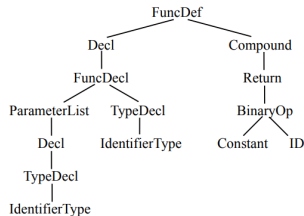
(Mou et al., 2014)

Идея:

- Вместо текста кода рассмотрим его синтаксическое дерево
- Обучаем Code2Vec, используя в качестве контекста сыновей в синтаксическом дереве
- Нормализуем вектора сыновей по размеру их поддерева

```
double doubles(double doublee){  
    return 2 * doublee;  
}
```

A C code snippet



The corresponding AST

Подходы к анализу кода





Использование метода

- Обучаем модель на корпусе кода фиксированного языка программирования (например, Python)
- В качестве embedding-ов вершин синтаксического дерева используем полученные векторные представления
- Используем RNN, как в случае текста, отдавая на вход полученные embedding-и в порядке обхода *dfs*-ом

Выводы

Результаты

- *сравнение с имеющимися решениями*
- Анализ того, какие текстовые представления работают лучше всего
- Анализ того, как улучшилась классификация после добавления учета содержания сниппетов кода

- ①  [Tian et al. \(2013\)](#)
Towards Predicting the Best Answers in Community-Based Question-Answering Services
- ②  [Gkotsis et al. \(2014\)](#)
It's all in the Content: State of the art Best Answer Prediction based on Discretisation of Shallow Linguistic Features
- ③  [Calefato et al. \(2016\)](#)
Moving to Stack Overflow: Best-Answer Prediction in Legacy Developer Forums
- ④  [Mou et al. \(2014\)](#)
Building Program Vector Representations for Deep Learning