Факултет по математика и информатика

Софийски университет „Свети Климент Охридски“

Курсов проект

Тема:

Transfer learning

Изготвили:
Димитър Керезов, Фак. № 61700
Стоян Тодоров, Фак. № 61675

Съдържание

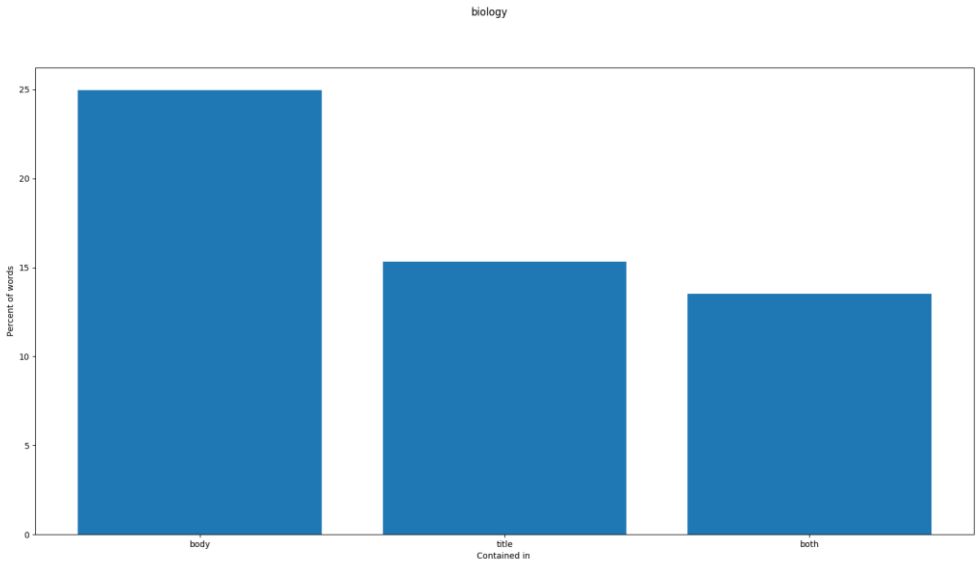
Задание

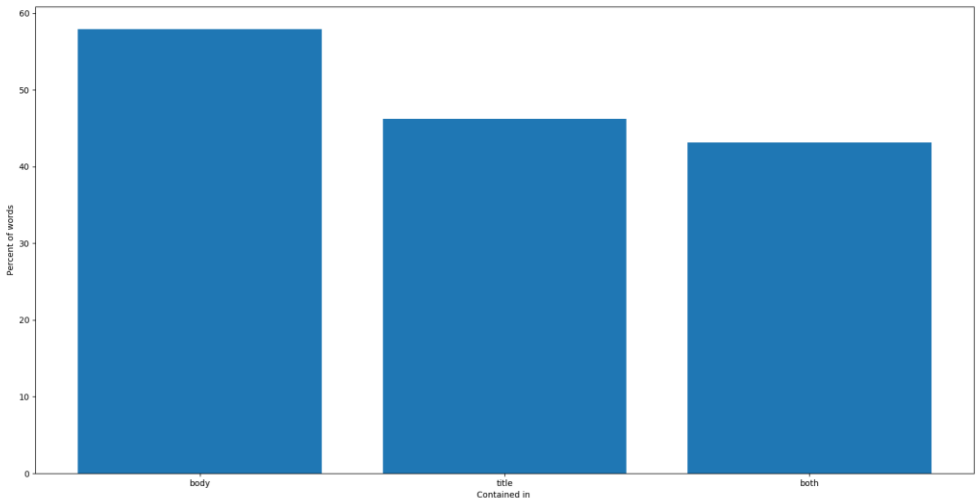
Дадени са заглавията, текстовете и таговете на въпроси от въпроси от stackexchange от 6 различни тематики – **биология, готварство, криптография, направи си сам, роботика и пътешествия**. Целта е да се представят предсказания за това какви биха били таговете на въпроси от областта на **физиката**. Целта е да се провери дали може да се добие знание за това как се правят тагове от дадените 6 области и това знание да се приложи в съвсем различна област.

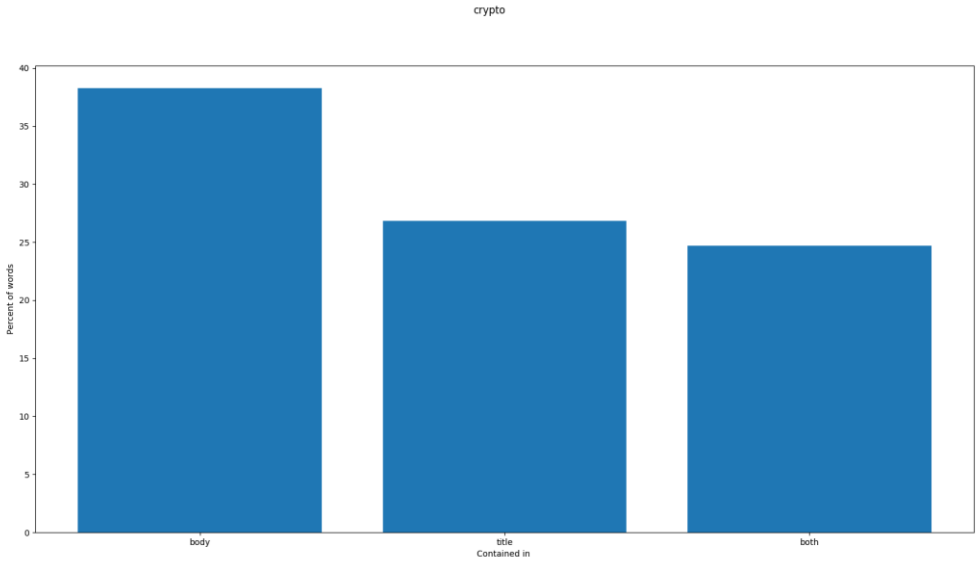
Заданието на тази задача е взето от сайта kaggle.com и може да се види на адрес https://www.kaggle.com/c/transfer-learning-on-stack-exchange-tags

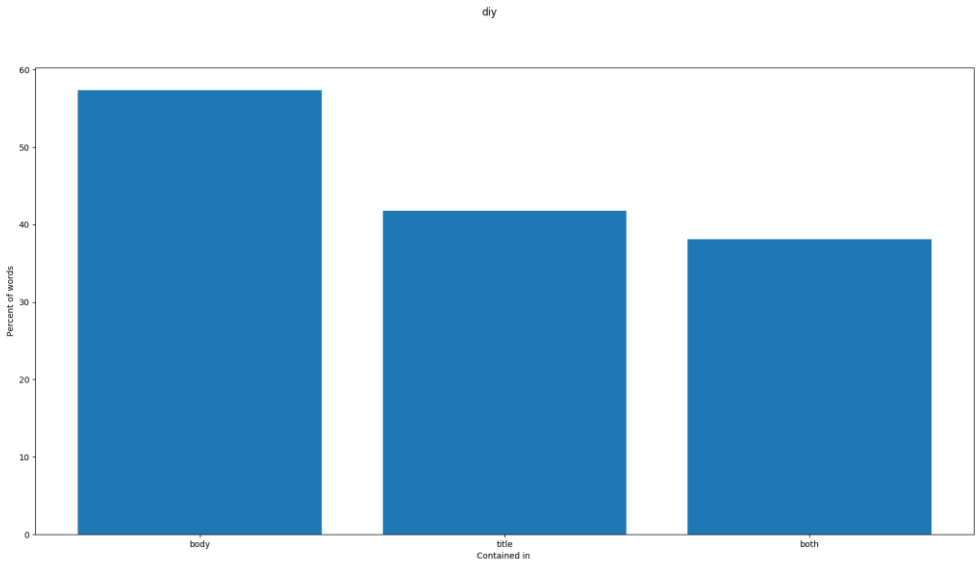
Анализ на проблема

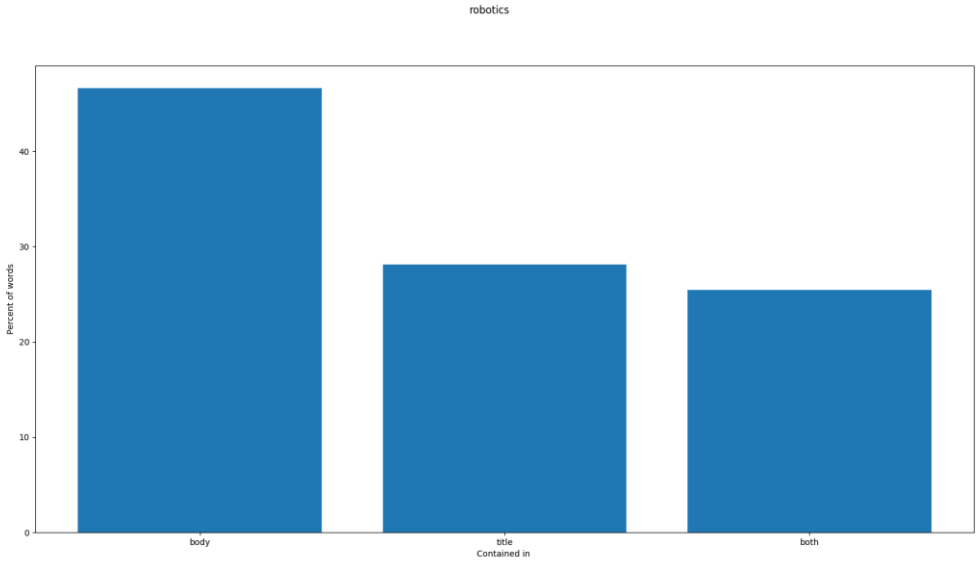
Започнахме с анализ на вече дадените тагове за 6-те теми. Анализирахме колко процента от тях се срещат в заглавието на съответния им въпрос, колко – в текста и колко – и в двете:

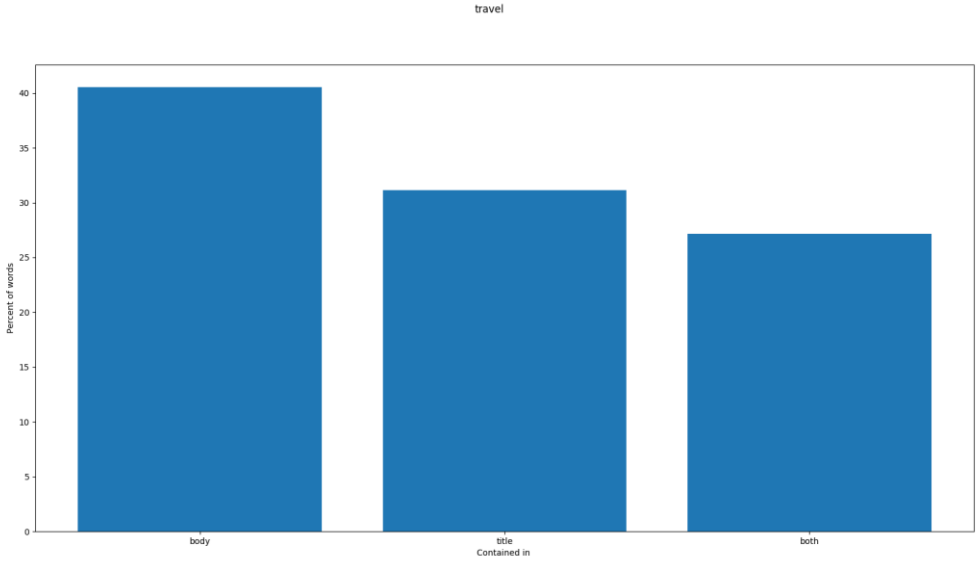





crypto
Percent of words
0
5
10
15
20
25
30
35
40
body
title
both
Contained in

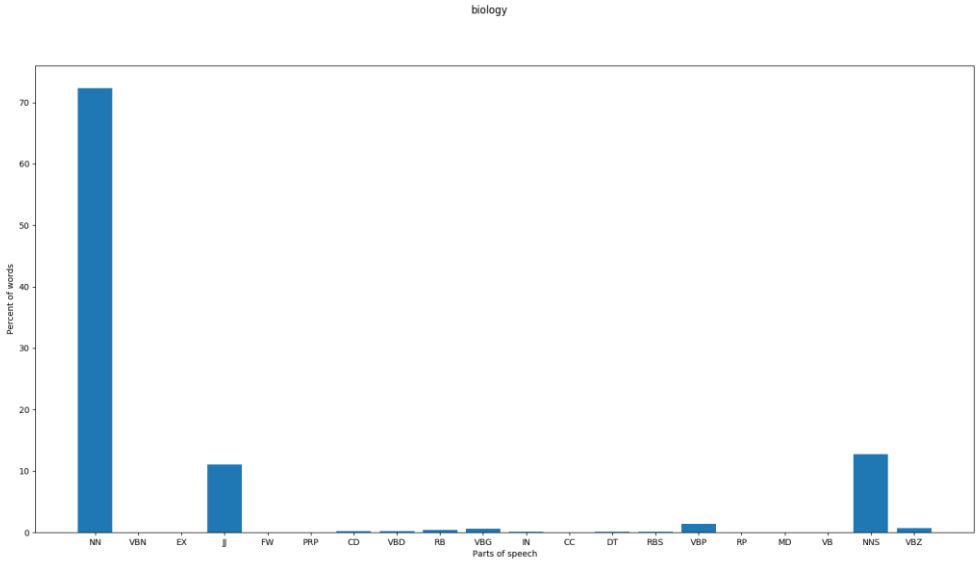

diy
Percent of words
0
10
20
30
40
50
60
body
title
both
Contained in

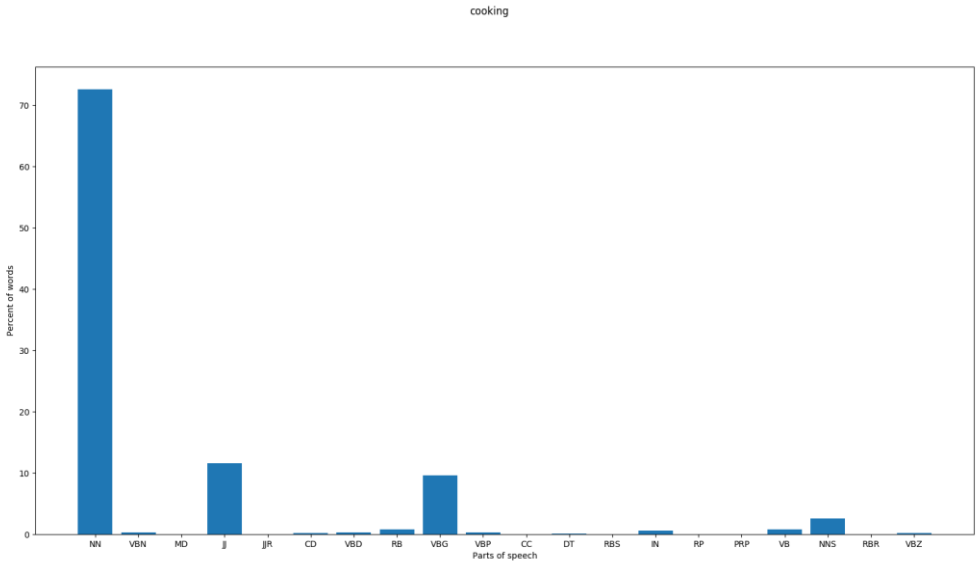




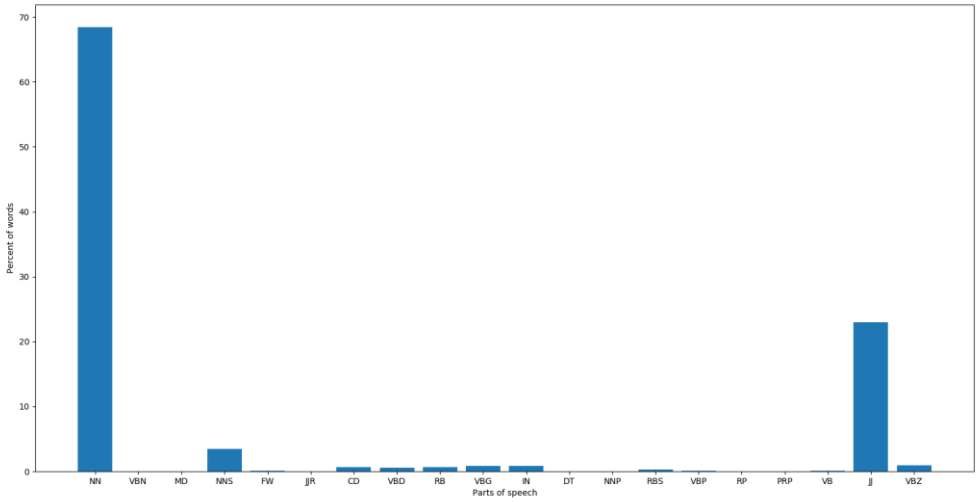
Оттук произлезе първия проблем – голяма част от думите, които са тагове не се съдържат нито в текста, нито в заглавието на въпросите.

Като втора стъпка решихме да видим каква част на речта е всеки един от таговете:

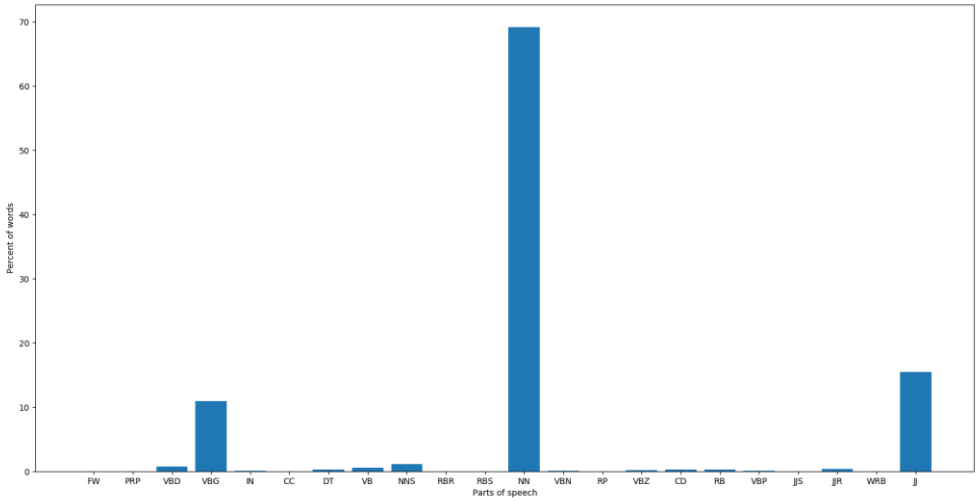


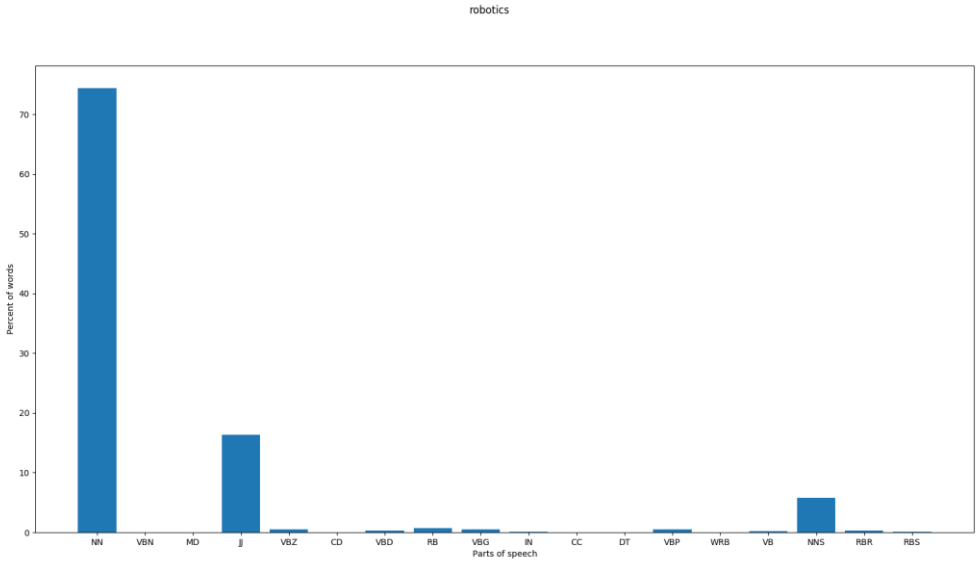


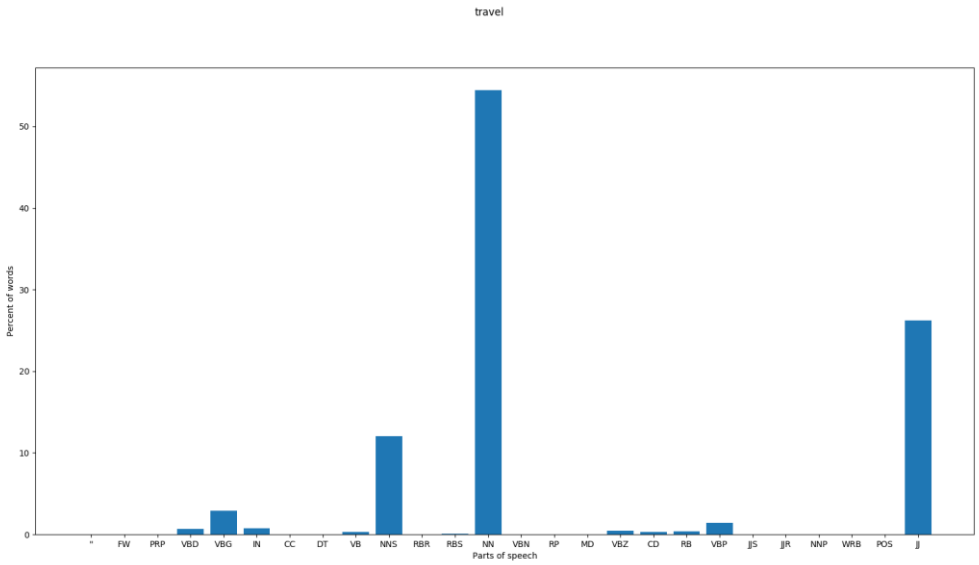
crypto


Percent of words
Parts of speech
NN
VBN
MD
NNS
FW
JJR
CD
VBD
RB
VBG
IN
DT
NNP
RBS
VBP
RP
PRP
VB
JJ
VBZ

diy

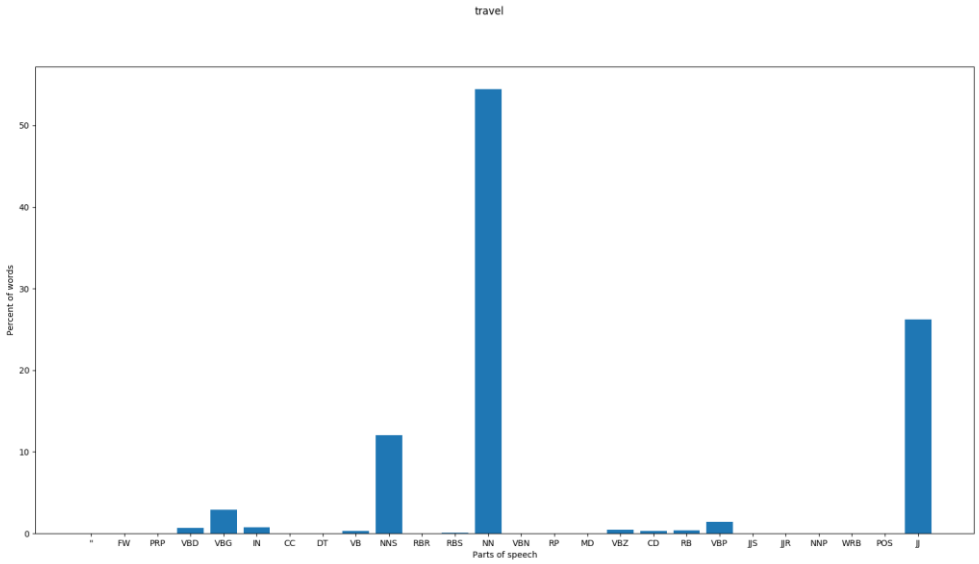

Percent of words
Parts of speech
FW
PRP
VBD
VBG
IN
CC
DT
VB
NNS
RBR
RBS
NN
VBN
RP
VBZ
CD
RB
VBP
JJS
JJR
WRB
JJ





Забелязахме, че повечето от думите, които са тагове спадат в категориите NN(обикновени съществителни имена), JJ(нестепенувани прилагателни имена), NNS(съществително име, множествено число) и VBG(глагол в ing форма).

В процеса на работа ни направи впечатление, че доста от темите имаха тагове, които са сложни, полуслято изписани думи. Проверихме колко точно от въпросите имат поне по един такъв таг:



След този предварителен анализ преминахме към реализация на решение.

Подходи към проблема

Решихме да използваме TFIDF алгоритъма като първа стъпка. Този алгоритъм съпоставя всяка дума със стойност за това колко е значима думата в даден контекст. Приложихме този алгоритъм и разбрахме стойностите на всеки от таговете в контекста на заглавието и текста на въпроса. Срещнахме проблеми с това, че алгоритъмът не беше приложим над думи, които не се срещат в заглавието или в текста на въпроса – тях просто игнорирахме. Взехме средното аритметично на всички стойности за всички тагове за всяка категория:

Категория	**Средно заглавие**	**Средно текст**
Билогия	0.300763	0.291047
Готварство	0.366773	0.373715
Криптография	0.269050	0.261347
Направи си сам	0.308642	0.322381
Роботика	0.249644	0.249683
Пътешествия	0.276647	0.276553
Средно	0.295253	0.295788

Спряхме се на контролна стойност **0.3**. Решихме да пуснем TFIDF за всяка дума от заглавието и текста на въпросите по физика и да вземем за тагове всяка дума, която е с по-висока стойност от контролната. Като допълнителна оптимизация, решихме да вземаме предвид само думите, които спадат към някоя от категориите NN, NNS, JJ или VBG.

За генериране на допълнителен набор от думи решихме за всяка дума, чиято стойност е по-ниска от контролната да вземем нейните хипероними и хипоними и да проверим тях – ако някой от тях има по-висока стойност от контролната да вземем него.

След TFIDF решихме да пробваме малко по-различен подход в лицето на алгоритъма LDA. С негова помощ взехме 5-те най-важни думи от текста на всеки въпрос, но те се оказаха силно неприложими, тъй като много малък процент от тях (под 5) се оказаха истински тагове.

Технологии

Проектът е реализиран изцяло на програмния език Python версия 3.5.2 с помощта на библиотеките nltk и sklearn. Графиките са изчертани със същия език за програмиране с помощта на библиотеката matplotlib

Бъдеще

За бъдещо развитие на проекта имаме още идеи, за които не стигна време за имплементация:

- За генериране на полусляти думи може да се приложи следния алгоритъм – от всеки две думи, чийто стойности попадат в диапазона между 0.2 и 0.3 спрямо TFIDF могат да се образуват двойки полусляти думи
- Да се помисли в насока как да се оценяват думи, които не са част от текста/заглавието на въпроса