Оглавление

Введение. Описание проблемы

Поисковая оптимизация - это процесс подготовки и организации контента на странице или на сайте для увеличения его потенциальной релевантности по определенным ключевым словам в определенной поисковой системе. SEO является аббревиатурой от Search Engine Optimization, что в переводе означает оптимизация под поисковые системы или просто поисковая оптимизация. Существует много компаний, предлагающих SEO-услуги по продвижению сайтов, но в результате некачественной работы таких компаний, в сети Интернет появляются новые сайты, не содержащие полезной для пользователя информации: встречаются сайты, на которых есть только призывы купить товар компании или воспользоваться ее услугами, также встречаются неграмотно написанные статьи, или тексты с большим количеством «воды».

Постановка задачи

Для улучшения качества поиска в сети Интернет необходимо анализировать сайты на качество текстов по определенным критериям и не допускать попадания рекламных и неграмотных текстов в выдачу поиска.

Поисковыми системами не производится анализ текстов на качество: периодически вводятся новые фильтры, но пока важна лишь уникальность контента, качество подачи информации, ее достоверность и т. д. Такой анализ не позволяет полностью исключить некачественные и бесполезные людям материалы и, соответственно, сайты, на которых таких материалов большинство.

Задача заключается в создании программного продукта, способного оценивать качество текстов.

Прежде всего, необходимо сформулировать критерии, по которым будут оцениваться тексты:

1. «Водность» текста.
2. Орфографическая корректность текста.
3. Грамматическая корректность текста.
4. Информативность текста.
5. «Рекламность» и тональность текста.

Данные критерии в рамках поставленной задачи удобно представлять в виде чисел, характеризующих качественные характеристики текста, например, от 0 до 1 (или 0-100%), где 0 – наихудший результат по конкретному критерию. Так как все тексты имеют разный размер, данные критерии должны быть в относительных единицах. Текст необходимо анализировать в обработанном виде, в котором все слова приведены к начальной форме, то есть, в именительном падеже и т.д., так как иначе будут ситуации, когда программа будет принимать, что два одинаковых слова в разных формах – разные. Например, слова «солнцам» и «солнцу» без

приведения к начальной форме «солнце» будут восприниматься программой как разные.

Необходимо также пояснить, как будут вычисляться оценки по каждому критерию в отдельности.

Определение оценок по критериям

«Водность» текста

Пользователи, которые занимаются созданием уникального контента, нередко сталкиваются с требованиями, чтобы процент «воды» в написанных текстах не превышал определенного значения.

В понимании SEO «вода» – это отношение стоп-слов к общему количеству слов на конкретной странице веб-сайта. То есть, чем больше в содержимом статей или записей ничего не значащих выражений и фраз, тем выше значение водности.

Этот критерий так же учитывают поисковые системы при ранжировании сайтов в выдаче результатов. Страница, на которой процент «воды» высок, очень быстро исчезает из ТОП выдачи. Помимо этого, переизбыток стоп-слов делает текст менее понятным и читаемым для посетителей сайта.

В рамках данного проекта, «водность» текста будет рассчитываться следующим образом:

$$V = \frac{W_v}{W_t},$$

где V- «водность» текста, W_v – число стоп-слов, W_t – общее число слов в тексте. Данный критерий может принимать значения от «0» до «1». Значение «0» означает, что в тексте отсутствуют стоп-слова, «1» – что текст полностью состоит из стоп-слов, но на практике, не бывает текстов, полностью состоящих из стоп-слов. Учитывая это, формулу для расчета оценки можно переписать следующим образом:

$$V=\begin{cases}\frac{2*W_v}{W_t}, if\ \frac{W_v}{W_t}<0,5\\ 1, if\ \frac{W_v}{W_t}\geq 0,5\end{cases}.$$

То есть, при количестве стоп-слов, большем, чем половина от общего числа слов, программа будет оценивать текст, как полностью состоящий из «воды». А при количестве стоп-слов, меньшем, чем пороговое значение, значение оценки будет равно удвоенному отношению числа стоп-слов к общему числу слов.

Орфографическая, синтаксическая корректность

Несмотря на то, что сейчас существует большое количество онлайн-сервисов, позволяющих произвести полную оценку грамотности текста, нередко попадаются страницы с большим количеством орфографических и синтаксических ошибок. Данный критерий имеет меньшую значимость, чем, например, информативность или рекламность, но, тем не менее, производя оценку качества текста, нельзя не учитывать грамотность текста.

Орфографическая правильность будет рассчитываться следующим образом:

$$orf_{err}=\frac{N_{or}}{W_t},$$

где *orf_err* – значение критерия «орфографическая правильность», N_{or} – число орфографических ошибок (неправильных слов), W_t – общее число слов в тексте. Данный критерий может принимать значение от «0» до «1», где «0» - значение, при котором все слова в тексте правильно написаны, при «1» все слова некорректны. Поскольку текстов, все слова которого неверно написаны, в сети Интернет не так много, формулу стоит дополнить, как и в случае с критерием «водность» текста.

$$orf_{err} = \begin{cases} \frac{2 * N_{or}}{W_t}, if \frac{N_{or}}{W_t} < 0{,}5 \\ 1, if \frac{N_{or}}{W_t} \geq 0{,}5 \end{cases}.$$

При количестве орфографических ошибок, большем, чем половина от общего числа слов, программа будет оценивать текст, как абсолютно неграмотный. А при количестве ошибок, меньшем, чем пороговое значение, значение оценки будет равно удвоенному отношению числа орфографических ошибок к общему числу слов.

Синтаксическая правильность будет рассчитываться следующим образом:

$$syn_{err} = \frac{N_{sr}}{W_t},$$

где *syn_err*- значение критерия «синтаксическая правильность», N_{sr} – число синтаксических ошибок, W_t – общее число слов в тексте. Аналогично критерию «орфографическая правильность», формула была дополнена:

$$syn_{err} = \begin{cases} \frac{2 * N_{sr}}{W_t}, if \frac{N_{sr}}{W_t} < 0{,}5 \\ 1, if \frac{N_{sr}}{W_t} \geq 0{,}5 \end{cases}.$$

Информативность

В сети Интернет существует масса страниц с грамотными текстами, без стоп-слов, но полностью состоящими из повторяющихся предложений. В таких текстах где-то вставляют призыв купить какой-то товар, или ключевую фразу, которая обеспечит сайту приток посетителей из поисковой системы. Но такие страницы – некачественные, они показывают негативное влияние поисковой оптимизации, производимой недобросовестными SEO

компаниями, на сеть Интернет. Кроме исключения подобных текстов, данный критерий так же покажет информативность и наличие смысловой нагрузки в тексте.

Оценка по данному критерию будет рассчитываться по следующей формуле:

$$inf = \frac{W_{var}}{W_t},$$

где inf - значение критерия «информативность», W_{var} – число различающихся слов, W_t – общее число слов в тексте. Возможные значения в диапазоне от «0» до «1». «0» соответствует тексту, который состоит из одного повторяющегося слова, «1» - тексту, в котором все слова различны. Учитывая, что предельное значение «0» встречается в текстах редко, формула была расширена:

$$inf = \begin{cases} \frac{W_{var}}{W_t}, if\ 0{,}3 < \frac{W_{var}}{W_t} < 0{,}8 \\ 1, if\ \frac{W_{var}}{W_t} \geq 0{,}8 \\ 0, if\ \frac{W_{var}}{W_t} \leq 0{,}3 \end{cases}.$$

При значениях оценки, меньших либо равных 0.3, приравнять к 0, при значениях, больших 0.8 – к 1. В интервале от 0.3 до 0.8 оставлять значение без изменений.

Тональность

Для определения уровня тональности нет точной формулы, связанной со словами, поэтому необходимо использовать другой способ вычисления оценки.

Для вычисления тональности, были выбраны искусственные нейронные сети (ИНС), поскольку данная задача – задача классификации, для решения которых распространено применение ИНС (обучение с учителем). В рамках задачи классификации необходимо будет подготовить базу данных для обучения и тестирования сети, а также определить сами классы для задачи. Возможные выходные значения нейросети: 0, 0.25, 0.5, 0.75, 1. Пояснения для данных значений:

Значение выхода	Объяснение	Пример
0	Негативная (ярко выраженная)	Ужасный фильм! Зря потратил на него свое время, нужно было сразу выкинуть в мусор.
0.25	Негативная	Мне не понравился фильм, который я смотрел на прошлой неделе, хотя я видел и хуже.
0.5	Нейтральная	Этот фильм будет в кинопрокате на следующей неделе.
0.75	Позитивная	Вчера я смотрел хороший фильм, могу его посоветовать.
1	Позитивная (ярко выраженная)	Лучший фильм, который я когда-либо видел! Музыка, актеры, сценарий – все просто прекрасно!

Сама по себе тональность не играет практически никакой роли в оценке качества текста: она усиливает рекламный эффект текста, если таковой имеется. То, как должны быть связаны тональность и «рекламность», будет определено на этапе тестирования проекта.

«Рекламность»

Аналогично тональности, вычисление данного критерия является задачей классификации. Для ее решения так же будет применяться искусственная нейронная сеть с обучением с учителем, что требует заранее определить возможные значения выхода ИНС и объяснить их:

Значение выхода	Объяснение
0	Не рекламный текст (до 5% рекламы)
0.25	По большей части – познавательный материал, но есть предложение о покупке (до 10% рекламы)

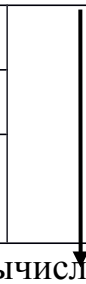
0.5	Есть и познавательный материал, и немало рекламной информации. (до 20% рекламы)
0.75	Рекламный текст, но несет немного полезной информации для пользователя (до 50% рекламы)
1	Полностью рекламный текст, почти никакой полезной информации, кроме описания продукта с лучшей стороны (больше 80% рекламы)

Общая оценка

Критерии оценки:

1. «Водность» текста.
2. Орфографическая корректность текста.
3. Грамматическая корректность текста.
4. Информативность текста.
5. Тональность текста.
6. «Рекламность» текста.

Точная формула общей оценки будет выведена на этапе тестирования программы. На данном этапе работы можно расположить критерии по их значимости:

«Рекламность»+тональность	
Информативность	
«Водность», орфографическая корректность, грамматическая корректность	

Стрелкой показано уменьшение значимости критерия в вычислении общей оценки текста.

«Рекламность» имеет самый большой вес в вычислении общей оценки текста, так как автор может быстро устранить грамматические ошибки, убрать лишние стоп-слова, увеличить информативность текста, но если текст был написан с целью продать какой-то товар, то это он изменить не сможет.

После критерия «рекламность» - информативность. Это связано с тем, что немаловажной является фильтрация страниц, заполненных одинаковыми

предложениями с ключевыми фразами, созданных для продвижения сайта в поисковой выдаче.

«Водность», орфографическая и синтаксическая корректность текста – показатели того, насколько внимательно писали текст, насколько автор разбирается в теме текста, поэтому вес данных критериев в вычислении общей оценки текста было решено принять примерно одинаковым и самым маленьким.

Язык программирования Python

Описание и обоснование выбора

Язык программирования Python является скриптовым (сценарным) языком. Скриптовый язык – язык программирования, разработанный для записи "сценариев", последовательностей операций, которые пользователь может выполнять на компьютере.

Был выбран именно Python, поскольку на нем высокая скорость разработки, что связано с простотой языка. В качестве примера различий в скорости разработки можно привести обмен значений двух переменных на языке Python и на языке c#:

Python	C#

a , b = b , a	a = a + b;
	b = a - b;
	a = a - b;

На Python для этого потребовалась всего лишь одна строка кода. Но основное различие между этими языка является в том, что в языке Python – динамическая типизация, что позволяет сильно сократить время разработки в случае работы со строками. В качестве примера, сравнение вывода числа на языках Python и Pascal:

Python	Pascal
int i=0 print(i)	int i=0; ShowMessage(IntToStr(i));

Различие в том, что в Python не требуется каждый раз писать функцию, конвертирующую численную переменную в строковую (IntToStr). Эта особенность Python как языка с динамической типизацией сильно повлияет на скорость разработки, поскольку в данном проекте будет много обработки строковых переменных.

Также, на скорость разработки влияет то, что язык Python – интерпретируемый: интерпретатор читает исходный текст программы по одной строке за раз, выполняет эту строку и только после этого переходит к следующей строке. А в компилируемых языках компилятор забирает часть времени разработки на себя, поскольку весь исходный код целиком преобразует в эквивалентную программу на низкоуровневом языке, близком машинному коду, и каждую компиляцию разработчик вынужден ждать, пока этот процесс закончится.

Библиотека PyMorphy2

Данная библиотека представляет собой морфологический анализатор, она находится в свободном доступе и имеет открытый исходный код, что позволяет всем пользователям вносить улучшения и модернизировать ее. В данной библиотеке используются словари OpenCorpora.

В результате анализа какого-либо слова пользователь может получить следующую информацию: полный орфографический разбор слова (часть речи, число, род, форма глагола, вид, время и др.), а также начальную форму слова.

Пример работы морфологического анализатора:

```
>>>morph.parse('стали')
[Parse(word='стали', tag=OpencorporaTag('VERB,perf,intr plur,past,indc'), normal_form='стать',
score=0.983766, methods_stack=((<DictionaryAnalyzer>, 'стали', 884, 4),)),

 Parse(word='стали', tag=OpencorporaTag('NOUN,inan,femn sing,gent'), normal_form='сталь',
score=0.003246, methods_stack=((<DictionaryAnalyzer>, 'стали', 12, 1),)),

 Parse(word='стали', tag=OpencorporaTag('NOUN,inan,femn sing,datv'), normal_form='сталь',
score=0.003246, methods_stack=((<DictionaryAnalyzer>, 'стали', 12, 2),)),

 Parse(word='стали', tag=OpencorporaTag('NOUN,inan,femn sing,loct'), normal_form='сталь',
score=0.003246, methods_stack=((<DictionaryAnalyzer>, 'стали', 12, 5),)),

 Parse(word='стали', tag=OpencorporaTag('NOUN,inan,femn plur,nomn'), normal_form='сталь',
score=0.003246, methods_stack=((<DictionaryAnalyzer>, 'стали', 12, 6),)),

 Parse(word='стали', tag=OpencorporaTag('NOUN,inan,femn plur,accs'), normal_form='сталь',
score=0.003246, methods_stack=((<DictionaryAnalyzer>, 'стали', 12, 9),))]
```

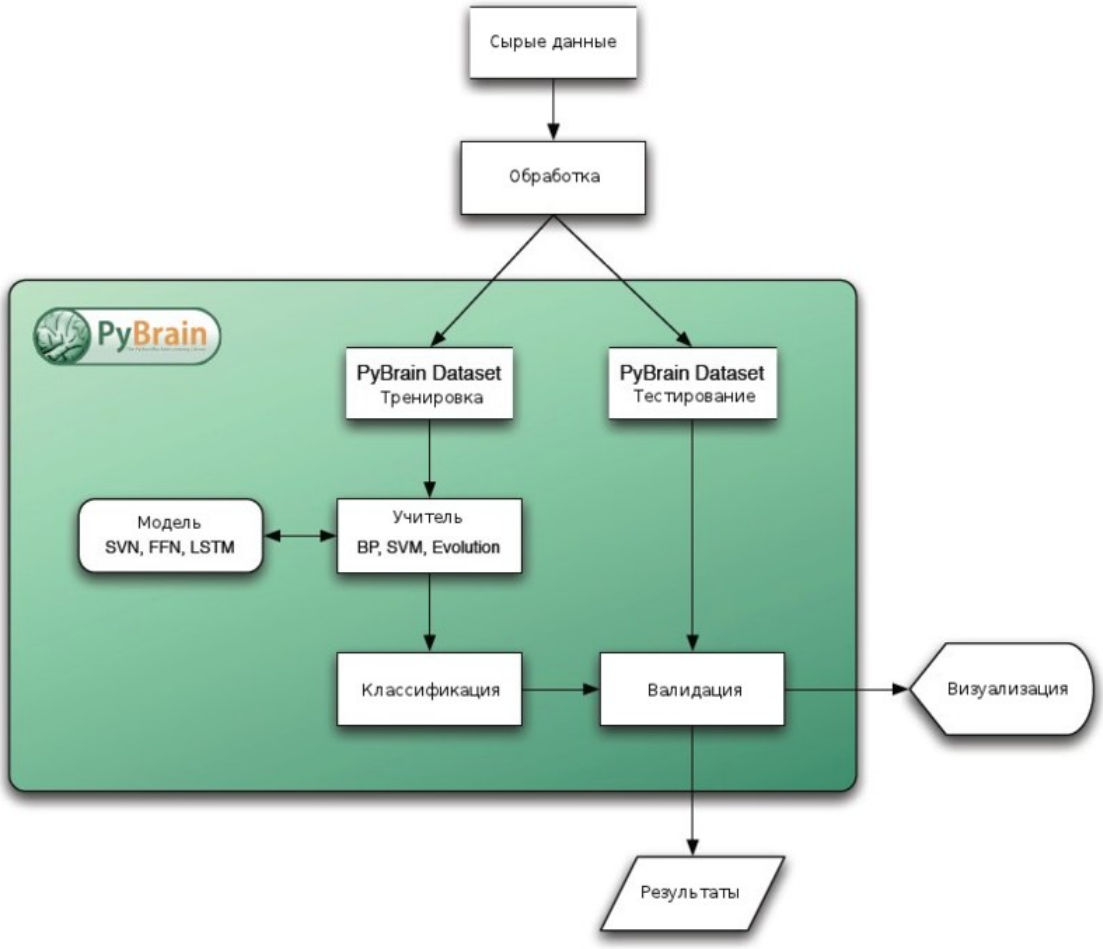
Команда parse выдает возможные варианты разбора слова списком, от более вероятного к менее. Вероятность хранится в словаре. Среди данных, которые выдает команда, для решения поставленных в проекте задач нужны normal_form и tag, а конкретно, часть речи (‘VERB’), которая доступна через команду tag.POS:

```
>>> import pymorphy2
>>> morph=pymorphy2.MorphAnalyzer()
>>> p=morph.parse('стали')[0]
>>> p.tag.POS
'VERB'
>>> p.normal_form
```

Библиотека PyBrain

PyBrain представляет собой модульную библиотеку, предназначенную для реализации различных алгоритмов машинного обучения на языке Python. Основной ее целью является предоставление гибких, простых в использовании, но в то же время мощных инструментов для реализации задач из области машинного обучения, тестирования и сравнения эффективности различных алгоритмов.

Общая структура процедуры использования PyBrain приведена на следующей схеме:



Данная библиотека позволяет реализовывать следующие искусственные нейронные сети с обучением с учителем, которое необходимо в рамках проекта:

- Алгоритмы обучения с учителем (Supervised Learning).
  - Метод обратного распространения ошибки (Back-Propagation)
  - R-Prop (Resilient propagation)
  - Support-Vector-Machines (интерфейс к сторонней библиотеке LIBSVM)
  - Evolino

В ходе работы над проектом будут протестированы разные алгоритмы обучения с учителем, проведен анализ результатов и выбран лучший.

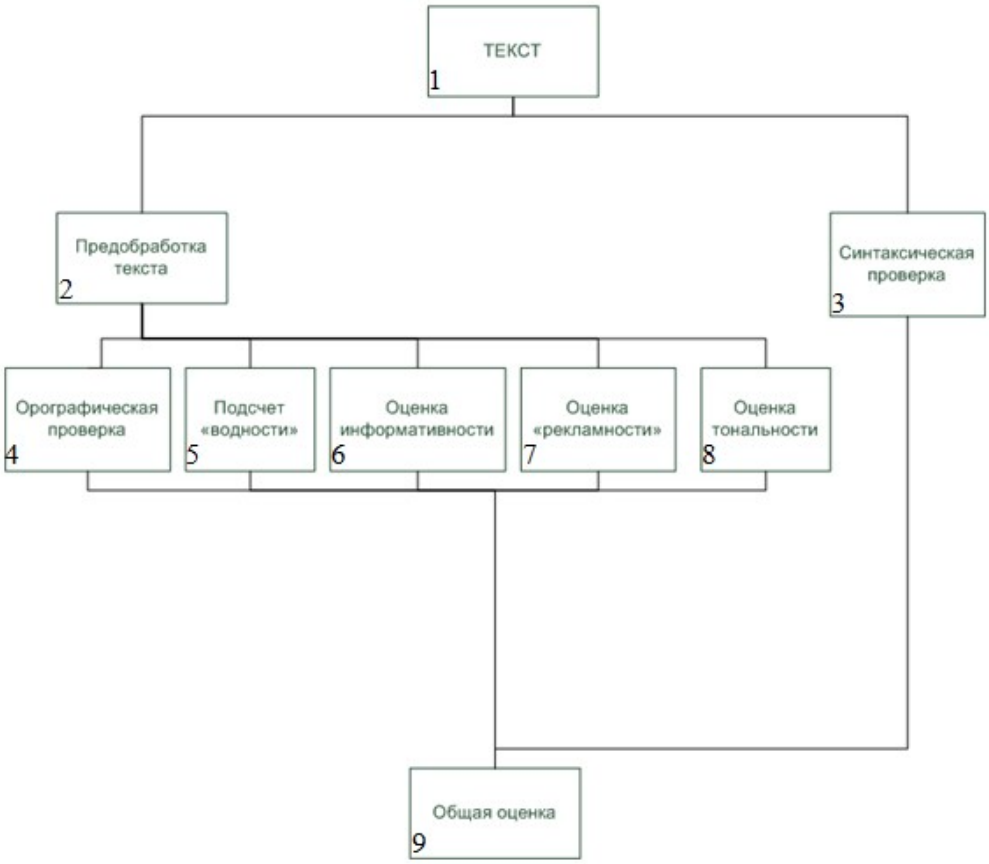
PyBrain оперирует сетевыми структурами, которые могут быть использованы для построения практически всех поддерживаемых библиотекой сложных алгоритмов. В качестве примера можно привести:

- Сети прямого распространения, включая Deep Belief Networks и Restricted Boltzmann Machines (RBM)
- Рекуррентные нейронные сети (Recurrent networks — RNN), включая архитектуру Long Short-Term Memory (LSTM)
- Multi-Dimensional Recurrent Networks (MDRNN)
- Сети Кохонена / Self-Organizing Maps
- Reservoirs
- Нейронная сеть Коско / Bidirectional networks [11]

Конечный выбор структуры сети так же будет осуществлен как результат тестирования нескольких видов с целью выбора наилучшего.

Алгоритм программы

Структура программы

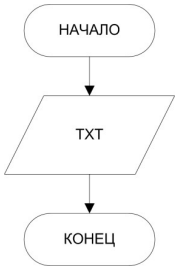


Описание блоков

Текст

Входные данные программы – файл с текстом, который требуется проанализировать. В этом блоке происходит считывание текста из файла.

Алгоритм:



Предобработка текста

В данном блоке происходит разбор текста на слова, а также приведение их к начальной форме с помощью библиотеки pymorphy2. Результатом работы данного блока является частотный словарь текста.

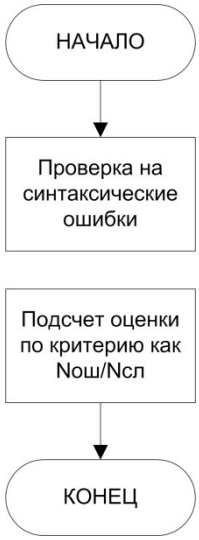
Алгоритм:

Синтаксическая проверка

На входе данного блока – текст. В этом блоке происходит анализ текста на наличие синтаксических ошибок с помощью библиотеки

LanguageTool. Результат работы блока – оценка синтаксической корректности текста.

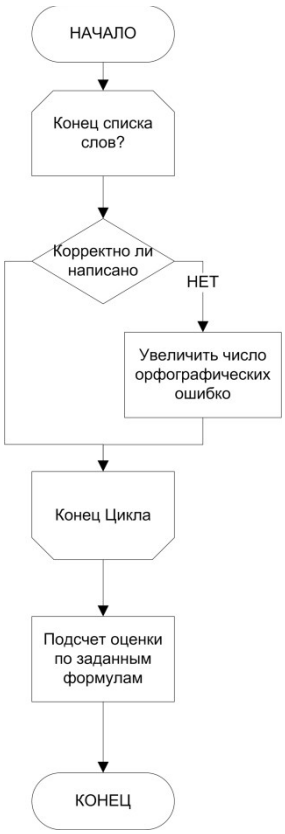
Алгоритм:



Орфографическая проверка

На входе данного блока – частотный словарь. В этом блоке происходит анализ текста на наличие орфографических ошибок с помощью библиотеки PyEnchant, которая осуществляет проверку правильности написания слов. Результат работы блока – оценка орфографической корректности текста.

Алгоритм:



Подсчет водности

На входе данного блока – частотный словарь. В этом блоке происходит анализ текста на «водность» с помощью библиотеки pymorphy2 (поиск определенных частей речи). Результат работы блока – оценка «водности» текста.

Алгоритм:

Оценка информативности

На входе данного блока – частотный словарь. В этом блоке происходит анализ текста на информативность. Результат работы блока – оценка информативности текста.

Алгоритм:

Оценка «рекламности»

На входе данного блока – частотный словарь, из которого для подачи в нейросеть в качестве униграмм отбираются все слова, кроме предлогов. В

этом блоке происходит анализ текста на «рекламность». Нейросеть должна быть предварительно обучена в соответствии с данными, приведенными в описании критерия. Результат работы блока – оценка «рекламности» текста.

Алгоритм:

Оценка тональности

На входе данного блока – частотный словарь, из которого для подачи в нейросеть в качестве униграмм отбираются все слова, кроме предлогов. В этом блоке происходит анализ текста на тональность. Нейросеть должна быть предварительно обучена в соответствии с данными, приведенными в описании критерия. Результат работы блока – оценка тональности текста.

Алгоритм аналогичен алгоритму оценки «рекламности»:

Общая оценка

На этом этапе суммируются оценки по всем критериям. Формула будет точнее определена в процессе тестирования для получения наилучших результатов. В данный момент примем:

$$\sum ¿ (V + orf_{err} + syn_{err} + inf + \left|((ton - 0.4) * rekl\right|)/5$$

В данной формуле показана связь тональности и «рекламности»: если текст нейтральный, то «рекламность» будет иметь меньший вес в конечной оценке, поскольку вероятность, что такой текст был написан для продажи чего-либо, уменьшается вместе с тональностью.

$(ton - 0.4)$ - здесь из оценки тональности вычитается 0.4, так как значения, которые принимает данная оценка: от «0» до «1», где «0» – негативная тональность, а «1» - позитивная. «-0.4» смещает шкалу, делая значения позитивной и негативной тональностей ближе по модулю значения друг к другу. Вычитается именно «0.4», а не «0.5», так как позитивная тональность оказывает более сильное влияние на читателя рекламы.

Заключение

В ходе выполнения данной работы были более точно определены критерии оценки качества текста, был написан примерный алгоритм вычисления оценки для каждого критерия.

Были выбраны язык программирования, основные библиотеки, позволяющие работать с искусственными нейронными сетями (PyBrain) и производить морфологический разбор слов (PyMorphy2).

Также, были выявлены некоторые формулы и значения, которые на данном этапе работы над проектом не является возможным точно установить, так как они требуют проведения ряда экспериментов с целью выявление зависимостей результата работы программы от них.

Список использованной литературы

1. «Кто использует Python?» [Электронный ресурс]. URL: http://python-3.ru/page/kto-ispolzuet-python
2. Обзор скриптовых языков программирования [Электронный ресурс]. URL: http://www.script-coding.com/Browse.html
3. Иванов И., Кокшаров С., Люстик А. SEO: Поисковая Оптимизация от А до Я [Текст]: в 3-х т. – 2016.
   Т. 1: Основы – 734 с.
   Т. 2: Средний уровень – 709 с.
   Т. 3: Продвинутый уровень – 927 с.
4. SEO блог «seoformat» [Электронный ресурс]. URL: http://seoformat.ru/istoriya_razvitiya/
5. SEO блог “Devaka” [Электронный ресурс]. URL: https://devaka.ru/articles/what-is-seo
6. Блог для вебмастеров (Яндекс) [Электронный ресурс]. URL: https://yandex.ru/blog/webmaster/20143
7. Интернет-энциклопедия «Машинное обучение» [Электронный ресурс]. URL: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение
8. Интернет-энциклопедия SEO [Электронный ресурс]. URL: http://wiki.rookee.ru/MatrixNet/
9. SEO блог [Электронный ресурс]. URL: http://www.apollo-8.ru/algoritm-matrixnet-yandex
10. Интернет-портал, посвященный Python [Электронный ресурс]. URL: http://pythonworld.ru/
11. Статья на портале habrahabr о PyBrain [Электронный ресурс]. URL: https://habrahabr.ru/post/148407/
12. Статья на портале habrahabr от создателя pymorphy [Электронный ресурс]. URL: https://habrahabr.ru/post/49421/