Лебедев Андрей, группа 324

Вариант 5
Отчет к заданию “Определение тональной окрашенности текста”

Постановка задачи:

1) Провести морфологический анализ словоформ текста. В контексте определения окрашенности необходимо лишь получение списка лемм с удалением стоп-слов.
2) Вычислить несколько общестатистических характеристик: общее число словоупотреблений, число уникальных лемм и т.п.
3) Определить тональную окрашенность текста, т.е. степень использования в нем оценочных слов и соотношение слов с положительной и отрицательной оценкой.
4) Вывести подсчитанные характеристики и оценки в удобной, обозримой форме.

Пояснения к решению каждой из подзадач

1) Морфологический анализ

Морфологический анализ, необходимый для определения тональности текста, должен состоять из лемматизации и фильтрации текста. Для реализации этого процесса служит функция text_preparing. Алгоритм ее работы:
а) Считывание текста из файла и токенизация. Токенизация осуществляется с помощью функции word_tokenize() библиотеки nltk.
б) Очистка текста от знаков препинания происходит “вручную”. Я создал список символов, которые могут возникнуть в списке word_tokens, после чего прошел по списку и в новый список words включил слова без пунктуации.
в) Лемматизация. Данный этап наиболее важен и производится использованием функции get_lemmas, которая принимает на вход список слов, а возвращает список лемм. В своем алгоритме она использует морфологический анализатор pymorphy3, осуществляющий парсинг для

каждого слова, из которого потом можно получить лемму с помощью normal_form.
г) Далее частично решается задача пункта 2 – вычисление общего числа словоупотреблений и числа уникальных лемм.
д) Следующий очень важный этап – очистка текста от стоп-слов. Они импортируются из корпуса nltk, формируясь в список, который я потом слегка расширяю. И похожим на очистку от знаков препинания алгоритмом убираем из текста мусорные слова, не влияющие на тональность текста.
е) Для наглядности список лемм вывел в отдельный файл lemmas.txt, о чем сообщил пользователю в консоли.

2) Общестатистические характеристики

При вычислении статистических характеристик решил не ограничиваться количеством словоупотреблений и уникальных лемм, добавив обзор на самые популярные леммы в тексте. Сделал это аналогично тому, как обсуждалось на семинарах – посредством класса FreqDist, позволяющим легко узнать абсолютную частоту каждой леммы в тексте.

3) Определение тональной окрашенности

В интернете описывается множество способов определения тональной окрашенности, но согласно рекомендациям преподавателей и постановке задачи, нужно соотнести количество положительных и негативных слов в тексте, а затем оценить результат.
Я решил воспользоваться тональным словарем русского языка kartaslov: https://github.com/dkulagin/kartaslov/tree/master/dataset/kartaslovsent. Он представляет из себя таблицу, в которой леммам сопоставляется тональная окраска, выраженная в численном виде. Так, у каждого слова есть значение value от -1 до 1, где -1 означает, что слово окрашено негативно, 1 – позитивно. Каждой лемме присваивается одна из категорий: Positive, Neutral или Negative. Интересно, что слово считается негативным, если value < 0, в то время как для позитивной категории необходимо условие value >= 0.55. Слово считается нейтральным, если 0 <= value < 0.55.
Алгоритм работы с тональным словарем состоит в следующем:
во-первых, используя средства модуля pandas, помещаю данные из csv таблицы в DataFrame. Затем я работаю только с теми словами, которым присвоены категории Negative или Positive, причина станет ясна позднее.

Далее для всех лемм текста, которые присутствуют в DataFrame, ищу их среднее арифметическое значение value. Таким образом мы находим среднее value всего текста, то есть тональную окрашенность текста.
Зачем было удалять из DataFrame нейтральные слова? Они имеют положительное значение value, из-за чего среднее арифметическое value для всего текста смещается ближе к 1, получаем позитивный текст там, где он таковым не является. Я решил, что нейтральные слова на то и нейтральные, что мы можем позволить себе считать, что они не влияют на окраску текста в целом.
Получив среднее арифметическое, нужно ответить на вопрос: как окрашен текст: позитивно, негативно или нейтрально? Путем многочисленных испытаний программы на разных текстах я пришел к следующим пороговым значениям среднего арифметического value:
value >= 0.45 – текст положительный;
value <= 0.2 – текст отрицательный;
0.2 < value < 0.45 – нейтральный.

Результаты экспериментов

Ниже привожу таблицу, содержащую результаты экспериментов, на основании которых я вывел пороговые значения value.

Содержание текста	Полученное value	Желаемый результат
Ф. М. Достоевский, "Преступление и наказание", диалог Сони Мармеладовой и Раскольникова'. Отрывок содержит мучительные размышления Раскольникова, поэтому негативный.	-0.256	Негативный текст
Ф. М. Достоевский, "Униженные и оскорбленные", отрывок. Текст - диалог Наташи и Алеши, где герои ссорятся.	0.084	Негативный текст
С. Есенин, "Черный человек" – известное лирическое стихотворение Есенина о страданиях.	0.045	Негативный текст
С. Аксаков, "Аленький цветочек" – добрая детская сказка.	0.455	Положительный текст
М. Горький, "На дне", действие 1 – знаменитая пьеса о безысходности.	0.012	Негативный текст
Негативный отзыв на ВМК с tabiturient.ru	0.078	Негативный текст
Позитивный отзыв на ВМК с tabiturient.ru	0.571	Положительный текст

Р. Брэдбери, "451 градус по Фаренгейту", диалог – знакомство главного героя с соседской девушкой.	0.397	Нейтральный текст
А. Приставкин, "Ночевала тучка золотая", отрывок о том, как бездомных детей поздравляют и дарят им подарки.	0.589	Положительный текст
И. Бунин, "Господин из Сан-Франциско" – отрывок об удовольствии, которое получает главный герой в путешествии, но под саркастической призмой Бунина, ведь все вокруг там - притворство. Таким образом, текст скорее о положительном, но из-за мрачных метафор, лживо-положительный.	0.388	Нейтрально - положительный текст
Позитивный рассказ о лете, сгенерированный в ChatGPT.	0.821	Положительный текст
Грустный рассказ об осени, сгенерированный в ChatGPT.	0.052	Негативный текст
Антуан де Сент-Экзюпери, “Маленький принц” – милый диалог главного героя с Лисом.	0.536	Положительный текст
"Алиса в Стране чудес" Льюиса Кэрролла, 1 глава. Повествование о том, как Алиса провалилась в нору.	0.212	Нейтральный текст
"Алиса в Стране чудес" Льюиса Кэрролла, 11 глава. События в суде, страх Алисы и злость Королевы.	0.024	Негативный текст
Приказ В. А. Садовничего “О деятельности Московского университета с 28 октября по 07 ноября 2021 года”	0.271	Нейтральный текст
Алан Милн “Винни-Пух и все-все-все”, глава 16 о подвиге Пятачка.	0.45	Положительно-нейтральный текст

Проблемы и способы их решения

В ходе тестирования программы я столкнулся с рядом проблем:

1. Слова, кажущиеся негативными, имеют положительный смысл в определенном контексте.
2. Биграмма “не добрый” будет иметь положительную окраску, что, очевидно, неправильно.

3. Пороговые значения value, описанные выше, являются неточными. Я вывел их по результатам многочисленных экспериментов, но, думаю, можно получить неправильные ответы на некоторых текстах.
4. Диапазон для присвоения категории “Нейтральный” достаточно широк, в него попадает как минимум половина обычных, не специально подобранных текстов.
5. Программа не справится с иронией и сарказмом.
6. Программа не сможет верно обработать неграмотно написанный текст.

Как решить эти проблемы:

1. Очевидно, стоит смотреть не на слова по отдельности, а хотя бы на триграммы или лучше n-граммы. Данный способ поможет немного учесть контекст, то есть решить проблемы 1 и 2 в списке выше. А используя нейронные сети и сохраняя контекст с начала абзаца или даже начала текста, можно решить эту проблему полноценно.
2. Чтобы уточнить значения value, стоит провести анализ не десятков экспериментов, а тысяч и более. И проанализировать такой объем данных можно только используя алгоритмы машинного обучения. Но тогда возникнет проблема с подготовкой данных для тестовой выборки.
3. Добавить корректировку текста, как это сделано, например, в поисковых системах.

Выше были описаны концептуальные способы решения проблем. Вот как анализирует тональную окрашенность текста русское NLP-сообщество:

1. Сверточные нейронные сети + Word2Vec: https://habr.com/ru/companies/vk/articles/417767/
2. Сверточные нейронные сети + Word2Vec + Fast.ai: https://habr.com/ru/articles/472988/
3. Библиотека Dostoevsky: https://github.com/bureaucratic-labs/dostoevsky

Прикладная задача

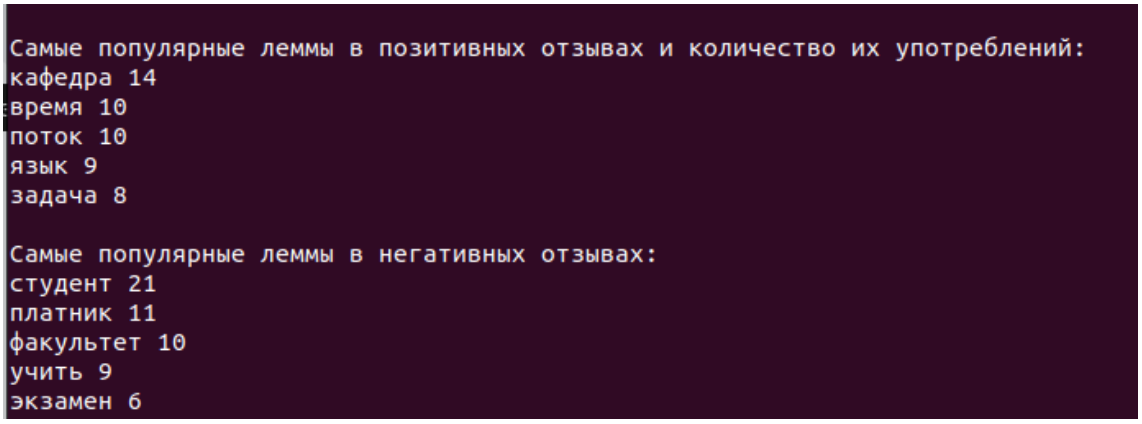
Примеры прикладных задач, решаемых анализом тональной окрашенности текста

1. Мониторинг общественного мнения: предприятия и организации могут использовать анализ тональной окрашенности для мониторинга

общественного мнения и отзывов о своих продуктах, услугах, бренде и об эффективности рекламы.
2. Прогнозирование финансовых рынков: финансовые аналитики могут использовать анализ тональной окрашенности новостных статей и социальных медиа для прогнозирования изменений на финансовых рынках. Эмоциональные реакции инвесторов могут влиять на цены акций и другие финансовые инструменты.
3. Фильтрация и сортировка контента: сервисы в социальных медиа и форумы могут использовать анализ тональной окрашенности для фильтрации спама и ненадежного контента.
4. Анализ политических и социальных событий: политологи и социологи могут применять анализ тональной окрашенности для изучения общественной реакции на политические решения, события и кампании.

В качестве прикладной задачи я выбрал из этого списка 1-й пункт и решил проанализировать общественные мнения об обучении на факультете ВМК. Для этого нашел отзывы (https://tabiturient.ru/vuzu/mgu/), загрузил их и провел следующую работу:

1. Имея написанную программу для анализа тональности текста, охарактеризовал каждый отзыв как положительный, негативный или нейтральный. Мои характеристики полностью совпали с теми, что есть на сайте.
2. Привел небольшую статистику: количество проанализированных отзывов, а также количество положительных и отрицательных среди них. Посчитал, что нейтральные отзывы являются лишь повторением и обобщением отзывов, имеющих категоричную оценку, и не стал с ними работать в дальнейшем.
3. Далее мне показался интересным способ анализа отзыва как на Яндекс Маркете: выделить наиболее употребляемые леммы в позитивных и негативных отзывах для получения общего представления о всех отзывах сразу. Получил следующие результаты:



Важно отметить, что среди наиболее популярных слов были такие, которые не несут смысла для анализа, например: учиться, вуз, семестр и другие. Поэтому я исключил их из топа.

4. Что нам дают эти результаты? С их помощью администрация факультета может сделать первичный анализ:
   1) Положительных отзывов меньше, чем негативных – это плохо.
   2) Студенты, удовлетворенные обучением на ВМК, акцентируют большое внимание на разделение на кафедры, значит, учиться 2 года общим дисциплинам, а затем выбрать что-то более узкое – хорошая стратегия.
   3) Такие студенты, вероятно, довольны временем, которое они тратят на обучение.
   4) Довольные студенты обеспокоены или были обеспокоены выбором потока при поступлении.
   5) Далее речь о языке, и тут 2 варианта – либо высказывалось отношение к изучению РЯКРа и английского языка, либо, что более вероятно, все та же обеспокоенность разделением на потоки, которое влечет изучение Паскаля либо Си.

   Я привел лишь примерные рассуждения, которые можно получить из этого анализа. Их можно развивать более глубоко, например: люди наиболее часто употребляют в тексте те слова, которые описывают вещи, наиболее беспокоящие их. Имею в виду то, что за основным посылом автора отзыва может встречаться дополнительный менее явный. Еще можно увеличить количество выводимых слов, но 5 мне показалось оптимальным.

Выводы

1. Морфологический анализ текста для решения задачи анализа тональной окрашенности включает в себя токенизацию, лемматизацию и удаление стоп-слов. Это позволяет получить список лемм, которые будут использованы для определения тональной окрашенности текста.
2. Для определения тональной окрашенности текста использован тональный словарь, в котором присвоены значения value от -1 до 1 каждой лемме. Среднее арифметическое значение value определяет тональность текста.
3. Пороговые значения для определения тональности установлены на основе многочисленных экспериментов, но могут быть неточными и требуют доработки.

4. Программа имеет ограничения, такие как невозможность учесть контекст, обработать иронию и сарказм, а также верно обработать неграмотно написанный текст.
5. Для улучшения анализа и устранения недостатков предлагается использовать методы машинного обучения, а также анализ контекста с использованием биграмм или n-грамм.
6. Ссылки на существующие NLP-решения и библиотеки (например, Dostoevsky) предоставлены для дальнейшего изучения и улучшения анализа тональности текста.

Моя работа представляет базовый подход к анализу тональной окрашенности текста, который хорошо справляется с задачей для моего уровня подготовки в АОТ, однако в целом все равно нуждается в улучшении и расширении методологии для более точных результатов. И тем не менее, с помощью написанной программы удалось решить полноценную интересную прикладную задачу.