УДК 623.332:629.27
DOI: 10.22213/2410-9304-2021-4-148-157

Emostemmer: эффективная программа для определения эмоций в русском языке с использованием N-грамм (эмоциограммы)

М. М. Аббаси, аспирант, Удмуртский государственный университет, Ижевск, Россия
А. П. Бельтюков, доктор физико-математических наук, профессор, Удмуртский государственный университет, Ижевск, Россия

Эмоции и анализ их выражения в текстах – тема растущего интереса в последние годы. Исследователи пытаются создать интеллектуальную машину, которая может не просто читать текст, но и определять его эмоциональный настрой. Полученные результаты могут быть использованы для подготовки машины к будущим предсказаниям эмоциональной ориентации текстов, их авторов и читателей. Данный анализ текста также может быть использован для получения обратной связи от людей о продукте или услуге, реакции на событие или на политику правительства и т. д. Он включает в себя синтаксический, а также семантический анализ текста. Синтаксический анализ состоит из определения слов, представляющих эмоции в тексте. Для его идентификации важную роль играет стеммер – основа или корень слова. Во многих языках романо-германской группы идентификация слов, представляющих эмоции, намного проще, чем в русском, поскольку одно слово представляет эмоцию независимо от грамматических форм и родов. В то время как для такого языка, как русский, где окончание слова, несущего эмоции, меняется в зависимости от рода, вида и др., анализ становится более сложным. Существуют разные методы определения эмоций в тексте.

В данной работе основное внимание уделяется выявлению эмоций из текста при ограничении сложности алгоритма требованием минимального объема памяти и времени. Нами была создана программа Emostemmer, которая представляет собой N-граммовый стеммер (в котором буквы из слов сгруппированы в последовательности из 2 букв, 3 букв… ..N букв, называемых N-граммами) для идентификации слов, которые представляют эмоции в тексте. Эффективность Emostemmer по сравнению с RuSentiLex определялась с помощью обучения и тестирования классификатора машины опорных векторов с обоими алгоритмами.

Ключевые слова: текст, эмоции, блог, общение, стемминг, анализ, матрица ошибок.

Введение

Эмоциональный анализ приобрел популярность в последние десятилетия. Он может быть выполнен с помощью анализов голоса, изображения, видео, сигналов и текста. Цель данной работы – изучить эмоциональный анализ текста. Это делается для его применения в прогнозировании будущих событий, идентификации людей, групп, в управлении обратной связью с клиентами по поводу продуктов или услуг, а также для улучшения взаимодействия человека с машиной, для создания машин, которые могут воспроизводить поведение и эмоции человека. В настоящее время существует огромный массив текстов, доступный в интернете в форме блогов, социальных сетей, опросов, тем для обсуждения и т. д., а также в автономном режиме в виде книг, журналов и других материалов. Такие тексты содержат очень важную информацию, которую можно обобщить, выявив эмоциональные компоненты текста. Тексты, доступные онлайн, находятся в неструктурированной форме, и осуществить их эмоциональный анализ – относительно сложная задача. Объем таких текстов увеличивается очень быстрыми темпами. Исследователи фокусируются на динамических текстах из социальных сетей и вебсайтов, чтобы выявить в них интересные тенденции.

Для идентификации эмоций текста обычно используются два основных подхода. Один включает в себя определение синтаксической ориентации текста, в то время как другой имеет дело с семантикой текста. Семантический подход отвечает за отнесение слова к соответствующей части речи, а затем идентификацию эмоций, выделяемых из него. Синтаксический же подход включает в себя изучение структуры текста. В его основе – определение слов, представляющих эмоции в тексте, их местоположения и частоты их появления.

Для этого используются разные алгоритмы для разных языков. Во многих языках романо-германской группы идентификация слов, представляющих эмоции, намного проще, чем в русском, поскольку одно слово представляет эмоцию независимо от грамматических форм и родов. В то время как для такого языка, как русский, где окончание слова, несущего эмоции, меняется в зависимости от рода, вида и др., ана-

лиз становится более сложным, как показано ниже (табл. 1).

Таблица 1. **Слова, которые представляют эмоции, и соответствующие родственные слова**
Table 1. **Words represents emotions and their corresponding root word**

Слова, которые представляют эмоции	Однокоренные слова
Радость	Рада, радостный, нарадоваться, радости, обрадовать, порадовать, на радостях, обрадованный
Любовь	Любить, любование, влюбляться, влюблен, себялюбие, налюбоваться, любимый, любя, любовный, любитель
Злость	Злой, зло, злоба, озлобленный, обозлиться, зловеще
Довольно	Довольствоваться, удовольствие, недовольно
Грусть	Грустный, грустить, грустно, взгрустнуть

Из приведенной выше таблицы видно, что слово с одинаковым значением может существовать в разных формах в тексте. В составе слова могут встречаться буквы, находящиеся до или после части слова, означающего эмоции. Наиболее распространенным способом является использование словарного метода, когда словарь состоит из сотен, а в некоторых случаях тысяч слов, и его необходимо поддерживать, чтобы идентифицировать все эмоции, выделяемые из текста. Это делает анализ сложным и трудоемким. Как известно, качество алгоритма измеряется временем и объемом памяти, которые требуются для его работы. В данном исследовании рассматриваются эти качественные показатели алгоритмов, а также точность определения эмоций, выделенных из текста с помощью предлагаемой программы Emostemmer.

Научные труды, связанные с данной работой

Основная цель применения того, что называется *стеммером,* состоит в том, чтобы свести различные грамматические формы и словоформы слова, такие как существительное, прилагательное, глагол, наречие и т. д., к его корневой форме. Первая статья об алгоритме стемминга была написана в 1979 году в компьютерной лаборатории в Англии и была опубликована в 1980 году как окончательный проектный доклад Ван Рейсбергена, С. Э. Робертсона и М. Ф. Портера [1]. Затем в том же году М. Ф. Портер опубликовал в своей работе суффиксные алгоритмы «зачистки» английского языка [2]. Ограничением морфологического варианта были производимые им не всегда настоящие слова. Оригинальный стеммер был написан на языке BCPL. Кровец Штеммер в 1993 году создал программу, преобразующую множественное число слова, встречающегося в тексте, в его единственное число, а также с помощью программы преобразовывал прошедшее время слова в настоящее [3]. В 1990 году Пейс Крис Д. предложил простой алгоритм для определения слов в тексте [4].

В 2003 году Уильям Б. Фрейкс и Кристофер Дж. Фокс предложили алгоритм стемминга. Целью этого алгоритма было определение слов, представляющих тему текста. Программа распознает такие слова, удаляя их суффиксы [5]. В 2005 году Микела Бачин, Никола Ферро и Массимо Мелуччи предложили стеммер, основанный на вероятностной модели, для извлечения важной информации из текста и последующей работы с ней [6]. В 2005 году Вибе вручную создал аннотацию слов, представляющих мнения и эмоции. Он аннотировал 10000 предложений, написанных в статьях, взятых из мировой прессы [7]. В 2007 году Пенг использовал стеммер в качестве средства поиска контекстной информации в сети. Он предположил, что стеммер может быть хорошим способом выявления важной информации из веб-текста [8]. В течение того же года Прасенджит Маджумдер предложил суффиксный алгоритм стеммера YASS («еще один суффикс-стриппер») [9].

В 2010 году Гиоргосом Адамом был предложен механизм анализа текста на греческом языке. Он предложил стеммер для пометки слов в тексте [10]. В том же 2010 году И. Фейнерер изучил проблему инверсии стемминга и определил ее причины (Over stemming, Under stemming) [11]. В 2011 году Джиолом Пэком предложен алгоритм стеммера для эффективного извлечения информации из текста. Стеммер был создан на модели, основанной на правилах системы rule based model, и были получены результаты для разных языков [12]. Затем к концу того же 2011 года Фернандез использовал неконтролируемую технику машинного обучения для создания программы изучения испанского языка [13]. В 2014 году был разработан араб-

ский стеммер, который строился с использованием синтаксических правил арабской грамматики. Необходимо отметить, что синтаксическая структура арабского языка отличается от английского языка, и стеммер должен иметь другие правила для поиска эмоциональных характеристик текста [14].

В 2014 году В. Данилова провела исследование методов извлечения событий из текста на нескольких языках [15]. В течение того же года С. Морал выполнил обзор алгоритмов, используемых для поиска информации на разных языках [16]. Методы и приемы, используемые для анализа эмоций в тексте на русском языке, были изучены в 2014 году Н. В. Лукачевичем и И. Четвёркиным. Они проанализировали и суммировали результаты семинаров ROMIP, проведенных в период с 2011 по 2012 г. [17]. В 2015 году С. Гарди и А. Мозай предложили стеммер, чтобы привести искаженные слова к их основной форме, и применили его для французского, английского и арабского языков. Было замечено, что на каждом языке стеммер ведет себя по-разному, в зависимости от уникальных синтаксических характеристик языка [18].

В 2015 году Томаш Бричин и Милослав Конопик предложили HPS (высокоточный инструмент Stemmer) для анализа текста. Они предположили, что «очистка» текста перед анализом может улучшить точность результатов стеммера [19]. В 2016 году Джасмит Сингх и Вишал Гупта провели всестороннее изучение текста. Они изучили подходы и определили современные проблемы, связанные с работой с текстом [20]. Аналогично, логические формулы используются для определения наиболее распространенных методов анализа эмоций, извлеченных из текста. Логические свойства и определение эмоциональных модальностей, логика эмоциональных оценок позволяют проводить эмоциональный анализ слов, обозначающих эмоции [21]. Недавно, в 2019 году, В. Несва и Бурсу выполнили функцию разметки речи посредством обучения без учителя. Они построили скрытую модель Маркова, в которой основы слов идентифицируются через модель скрытых состояний. Также изучалось влияние различных эмоциональных связей во всем тексте [22].

Выясняется, что большинство ученых работает над правилами для преобразования времен в их более простые формы, такие как going, gone, пытаются вернуть к первоначальной форме go. Точно так же большинство стеммеров имеет большую зависимость от языка и написаны были для английского языка. В связи с этим они хорошо работают только с языками, имеющими сходный синтаксис. Интересным вкладом стало создание языка Snowball для написания стеммеров [23].

В данной работе рассматривается создание основ определенной программы для русского языка с целью эффективного выявления эмоций из текста. В данной статье предложен алгоритм стеммера для идентификации эмоций из текста, написанного на русском языке. Это первый в своем роде алгоритм стеммера, разработанный для идентификации эмоций и их анализа на русском языке.

Эмоциональные конструкты

Изначально математические модели выражения эмоций в словах применяем в эмоциональном анализе текстов только для рассмотрения слов как последовательностей букв. Не учитывается рукописная, устная речь, шрифтовое и иное графическое оформление текста.

Предполагается, что моделью комплекса эмоций, выраженных в словах, является элемент некоторого множества Ex – множества *эмоциональных выражений,* и далее считаем, что зафиксировано некоторое такое множество Ex. На практике каждый его элемент отражает характер и силу различных эмоций, выраженных в слове. Кроме этого, анализируются эмоции, выраженные не только в словах, но и в более крупных единицах текста.

Функции эмоциональных выражений

Для того чтобы промоделировать комплекс эмоций, который выражает слово (возможно, даже зависимость этого комплекса от некоторых характеристик контекста, в котором это слово появилось), предполагаем, что имеется некоторая *функция эмоционального выражения* Fex, которая отображает множество слов рассматриваемого языка L во множество эмоциональных выражений Ex:

$$Fex: L \to Ex.$$

При этом лексика языка понимается как потенциальная – расширяющаяся с развитием языка. Интерес вызывает зависимость выражения этих эмоций от характеристик контекста. Это находит отражение в соответствующем усложнении структур объектов множества эмоциональных выражений Ex.

Алфавит языка обозначим буквой A. Так что здесь L – множество некоторых слов в алфавите A. Для того чтобы не решать слишком сложную задачу построения таблиц функции Fex для быстро расширяющегося и меняющегося множест-

ва L, рассмотрим способ, которым можно определять, какую роль играют те или иные части слов в определении значений функции эмоционального выражения этих слов.

Эмоциональное наполнение префиксов слов

Поскольку слова во время чтения воспринимаются человеком (субъектом, получающим эмоции) в порядке следования их элементов (букв), то наиболее естественным подходом к эмоциональному анализу слов является автоматный подход, сущность которого заключается в попытке разделения эмоционального анализа по частям слова в порядке его чтения. Здесь рассматриваются линейные текстовые языки с относительно небольшим числом букв. Например, таким языком является традиционная текстовая форма выражения русского языка. Тогда как, например, традиционная текстовая форма выражения китайского языка не рассматривается.

Для начала рассматриваем всевозможные разбиения слова w языка в порядке его написания и чтения на две части:

$$w = p\,s,$$

где p условно называем *префиксом*, а s – *суффиксом* этого разбиения.

Рассмотрим функцию *эмоционального выражения префикса* слов *Fpex* следующим образом. Пусть потенциальный префикс p – слово в алфавите A. Тогда значение $Fpex(p)$ – это отображение, которое каждому слову s в алфавите A такому, что ps – слово рассматриваемого языка (из L), ставит в соответствие значение $Fex(ps)$.

Эмоциональное наполнение инфиксов

Для того чтобы отражать вклад каждой части i слов в эмоциональные выражения всех слов, введем специальную функцию *Fiex*. Значение $Fiex(i)$ – это отображение, которое ставит в соответствие паре слов p и s в алфавите A, таких, что pis – слово рассматриваемого языка (из L), значение $Fex(pis)$. Слово i далее обобщенно называем *инфиксом*. Инфикс может быть основой слова (стемом), корнем или какой-либо другой частью, которую имеет смысл рассматривать при анализе эмоционального содержания слов.

Эмоциональный вклад инфикса i, кроме того, при автоматическом подходе может быть выражен как функция, выдающая по потенциальному префиксу p (слову в алфавите A) эмоциональный вклад его продолжения pi тоже как префикса – $Fpex(pi)$. Это выражается специальным оператором *Ftex*. Оператор *Ftex* выдает к слову i такую функцию $f=Ftex(i)$: что

$$f(p) = Fpex(pi).$$

Связь между функцией *Fiex* и оператором *Ftex* выражается следующей теоремой.

Теорема 1. Каковы бы ни были слова i, p и s в алфавите A, если слово pis принадлежит множеству L, то

$$Ftex(i)(p)(s) = Fiex(i)(p,s).$$

Доказательство теоремы нетрудно получить из приведенных выше определений функций *Ftex*, *Fiex* и *Fpex*.

Таким образом, основная задача сокращения объема работ при эмоциональном анализе текстов на развивающемся и расширяющемся языке состоит в поиске подходящих инфиксов (стемов), для которых алгоритм вычисления *Ftex* выражается с приемлемой вычислительной (временной и зонной – *Time*, *Space*) и описательной (объем программы) сложностью.

Разрабатываемая методология

Программа Emostemmer работает, стремясь использовать минимальные время и объем памяти с максимальной точностью. В основе программы имеется словарь не менее чем на 164 слова, представляющих простые формы или части слов, выражающих эмоции на русском языке. Алгоритм работает на специальных N-граммах, чтобы идентифицировать слова в разных формах; N-грамма для выявления эмоциональной переносимости будет называться эмоциограммой. Эмоциограмма – это список необходимых частей слов, передающих эмоции в тексте.

Эмоциограмма из n элементов может быть представлена в виде

$$D_{dict} = \left[DW_1, DW_2, DW_3 \ldots DW_n\right],$$

где DW_n – n-й элемент эмоциограммы.

Например, элементы эмоциограмм могут соотноситься со словами в соответствии со следующей табл. 2.

Таблица 2. **Эмоциограммы для выявления соответствующих эмоций из текста**
Table 2. **Emotiograms for identifying the corresponding emotions from the text**

Эмоциограмма	Эмоции
Везу́ч	Везу́чий, везу́чая, везу́чее, везу́чие, везу́чего, везу́чей, везу́чего, везу́чих, везу́чему везу́чим, везу́чую, везучими, везуч, везуча, везуче, везу́чи
Депре́сси	Депре́ссия, депре́ссии, депре́ссий, депре́ссиям, депре́ссию, депре́ссией, депре́ссиях, депрессиями
Гне́в	Гне́в, гне́вный, гне́ва, гне́вов, гне́ву, гне́вам,гне́вом,гне́вами, в гне́ве,
Краси́в	Краси́вый, краси́вая, краси́вое, краси́вые, краси́вого, краси́вой, краси́вых, краси́вому, краси́вому, краси́вым, краси́вую, краси́вое, краси́вого, краси́вых, краси́выми, краси́вом, краси́в, красива, красиво, краси́вы, краси́вее, покраси́вее, краси́вей, покраси́вей

Элементы эмоциограмм не всегда находятся в начале соответствующего им слова. Они могут размещаться в середине или даже в конце слова.

Описание этапов работы алгоритма Emostemmer

Для эксперимента мы использовали загруженный текстовый файл с разных сайтов и интернет-блогов на разные темы и применили к нему следующие операции:

$$T_{Text} = [Text]$$

(получение начального текста),

$$Z_{extra} = \{.,;:\%\ @\&!\$\}$$

(задание множества пропускаемых литер, символы),

$$T_{TextL} = f_{low}(T_{Text})$$

(функция f_{low} осуществляет перевод текста в строчные буквы),

$$D_{dict} = \{DW_1, \ldots DW_{164}\}$$

(задание начального списка эмоциограмм слов, выражающих эмоции),

$$T_c = f(Z_{extra})[T_{TextL}]$$

(оператор $f(Z_{extra})[T_{TextL}]$ удаляет символы Z_{extra} из текста, такие как русские «шумовые» слова),

$$W_s = f_{split}(T_c)$$

(функция f_{split} разбивает текст на слова выполняя токенизацию текста),

$$L_s = f'_{split}(W_s)$$

(функция f'_{split} представляет каждое слово как цепочку букв),

$$N_s = f_{\text{Эмоциограмма}}(L_s)$$

(функция $f_{\text{Эмоциограмма}}$ находит *N*-граммы, входящие в каждое слово, N = 3, 4, 5),

$$E_T = f_{comp}(N_s, D_{dict})$$

(функция f_{comp} ищет Эмоциограммы в словаре Emostemmer, E_T – цепочка чисел вхождений эмоциограммы каждого слова в словарь),

$$F = \sum_{i=1}^{n} E_T[i],\ n = length(E_T)$$

(определение общего числа вхождений эмоциограммы слов текста в словарь).

По результатам использования вышеприведенного алгоритма можно сделать вывод, что идентификация эмоций из текста им более эффективна по сравнению с существующими методами словаря, такими как RuSentiLex.

Эксперимент и сравнение производительности методов

Для проведения эксперимента с использованием Emostemmer для русскоязычного текста был загружен текстовый файл с разных сайтов и интернет-блогов на разные темы, например: образовательные организации. Одной из таких организаций был Удмуртский государственный университет (Ижевск), и отзывы собирались из разных блогов (https://udmurt.media/articles/obshchestvo/67638/, https://edunews.ru/universities-base/pfo/izhevsk/udsu.html, https://otzovik.com/review_8269223.html, https://studika.ru/izhevsk/udgu/otzyvi и др.) и в виде анкет студентов. Полученный текстовый файл содержит текст вместе с эмоциями на русском языке.

Для идентификации эмоций из текстового файла и классификации их использовались два разных источника русских лексиконов эмоции: эмоциограммы и RuSentiLex. Русский лексикон

эмоции RuSentiLex, созданный Н. В. Лукашевич и А. В. Левчиком [24–25], представляет собой сборник слов и фраз, отражающих эмоции в русском языке. Он состоит из 12 тысяч слов, которые используются для обозначения эмоций. Сборник классифицирует слова-эмоции как положительные или отрицательные. Источник RuSentiLex был получен с сайта (http://www.labinform.ru/pub/rusentilex). Он был загружен и использован для эксперимента, чтобы идентифицировать эмоции из текстового документа, параллельно с помощью программы Emostemmer.

Для Emostemmer список самых распространенных эмоций в русском языке был загружен с сайта психолога Петра Зарубина (https://peterzarubin.ru/spisok-chuvstv-i-emotsij) [26]. Список содержит 164 эмоции на русском языке. Эксперты создали основы (эмоциограммы) для этих 164 эмоций для Emostemmer. Было отмечено, что эмоциограммы, созданные для каждой эмоции, способны идентифицировать все возможные формы их выражения. Эксперимент был выполнен на Emostemmer, который состоит из 164 эмоциограмм. Для сравнения – RuSentiLex состоит из 12000 слов, представляющих эмоции.

Изначально программа проходила испытания на 12000 слов, представляющих эмоции в RuSentiLex вместе с их категориями. Затем программа была протестирована с использованием текстового файла, чтобы идентифицировать и классифицировать эмоции из текстового файла в их соответствующей группе на основе обучения алгоритму. Результаты идентификации и классификации представлены ниже в матрице ошибок (Confusion Matrix). Также программа обучалась с эмоциограммами, с их положительными и отрицательными категориями. Текстовый файл был обработан с использованием алгоритма Emostemmer, чтобы идентифицировать эмоции и классифицировать их в соответствующие положительные или отрицательные группы.

Матрица ошибок (Confusion Matrix) указывает на эффективность классификации алгоритма. При этом используются показатели, такие как Accuracy, Recall (полнота), Precision (точность) и F-measure (в общем случае), чтобы определить эффективность классификации алгоритма. На рис. 1 сгенерирована матрица ошибок для каждого алгоритма.

Рис. 1. Матрица ошибок для Emostemmer и RuSentiLex
Fig. 1. Confusion Matrix for Emostemmer and RuSentiLex

TP (True Positive) и TN (True Negative) уточняют, что значения истины, определенные матрицей ошибок, являются актуальными, тогда как FN (False Negative) и FP (False Positive) детализируют ошибки или неправильную классификацию, сделанные матрицей ошибок во время выявления и классификации эмоций. Было установлено, что алгоритм RuSentiLex не смог распознать все эмоции в тексте и пропустил некоторые из них, которые не были упомянуты в его словаре.

Результаты матрицы ошибок были использованы для определения истинности значений, выдаваемых алгоритмами Emostemmer и RuSentiLex, и подробно описаны ниже в табл. 3.

Таблица 3. **Матрица ошибок для Emostemmer и RuSentiLex**
Table 3. **Confusion Matrix for Emostemmer and RuSentiLex**

Algorithm	Accuracy	Recall	Precision	F-Measure
RuSentiLex	85,81 %	91,07 %	89,81 %	90,39 %
Emostemmer	93,55 %	95,4 %	95,4 %	95,00 %

Из анализа матрицы ошибок (Confusion Matrix) видна более высокая точность результатов программы Emostemmer. Матрица показала более высокую производительность и эффективность алгоритма Emostemmer для идентификации и классификации эмоций из текста по сравнению с RuSentiLex.

Также было отмечено, что время выполнения RuSentiLex выше, чем Emostemmer. Словарь RuSentiLex состоит из 12000 слов, и программа сравнивает все имеющиеся в его словаре слова с каждым словом из текста для идентификации и классификации эмоции из анализируемого текста до положительной или отрицательной категории. Для расчета времени выполнения каждого алгоритма в секундах текст для эксперимента был разбит на разные файлы с разным количеством слов. Программы Emostemmer и RuSentiLex выполнялись с использованием Software Matlab на компьютере Intel Core i5 7-го поколения. Полученные результаты представлены ниже в виде графика. Уменьшение времени выполнения задания Emostemmer по сравнению с RuSentiLex в среднем составляет 65 %. Полученные результаты представлены ниже в виде графика.

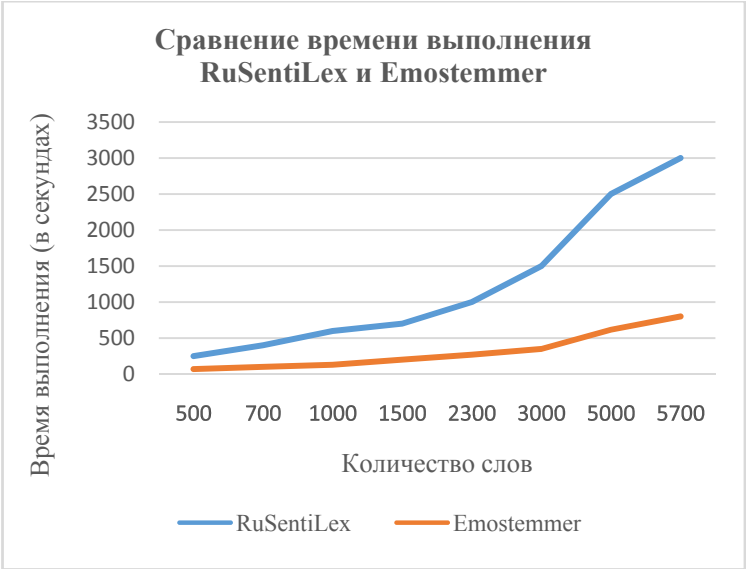

Рис. 2. Сравнение времени выполнения RuSentiLex и Emostemmer
Fig. 2. Comparison between execution time of RuSentiLex and Emostemmer

Из приведенного графика видно, что время выполнения анализа увеличивается с увеличением размера текста. Заметная разница наблюдалась между временем выполнения программ Emostemmer и RuSentiLex с начала эксперимента с минимальным размером текстового файла. Это произошло из-за сложности RuSentiLex, во многих случаях размер RuSentiLex был больше размера самого текстового файла, используемого для анализа.

Заключение

В данной статье были изучены и проанализированы механизмы, используемые для анализа эмоций, выделенных из текста. При этом основное внимание было уделено стеммерам и их работе с русскоязычным текстом. Основная цель построения стеммера состоит в том, чтобы свести различные формы (словоформы) слова, такие как его существительное, прилагательное, глагол, наречие и т. д., к его корневой форме. В работе был использован Emostemmer для идентификации эмоций, несущих слова из текста. До сих пор для русского языка использованию стеммера для идентификации эмоций не уделялось большого внимания. При сравнении производительности Emostemmer с RuSentiLex выяснилось, что Emostemmer превзошел метод на основе RuSentiLex для идентификации эмо-

ций из текста. Точность результатов Emostemmer также оказалась выше, чем точность RuSentiLex-метода. Время выполнения анализа Emostemmer заметно меньше времени выполнения RuSentiLex. Emostemmer выполняет вычисления для идентификации эмоций из текста более просто и разумно. В будущем мы планируем использовать его как часть нашего проекта для улучшения взаимодействия человека с машиной посредством текста.

Библиографические ссылки

1. Rijsbergen J., Robertson C. J., Stephen E., (1946) & Porter, Martin F. (1980). New models in probabilistic information retrieval // British Library Research and Development Dept., [London]. No. 5587.
2. Porter M.F. An algorithm for suffix stripping (1980). Emerald Publishing, Program 1 14 (3), 130-137.
3. Krovetz R (2000). Viewing morphology as an inference process // Artificial Intelligence Journal, Q1 SJR 1.01. 118(1), 277-294.
4. Paice C. D (1990). Another Stemmer // ACM SIGIR Forum, 24(3), 56-61.
5. William B. Frakes , Christopher J (2003). Fox. Strength and similarity of affix removal stemming algorithms // ACM SIGIR Forum, 37(1), 26-30.
6. Bacchin M., Ferro N., Lucci M (2005). A probabilistic model for stemmer generation // Information Processing and Management 41(1), 121-137.
7. Wiebe, J., Wilson T., Cardie C (2005). Annotating expressions of opinions and emotions in language // Language Resources and Evaluation 39 (2), 165-210.
8. Peng, F., Ahmed, N., Li, X., Lu, Y (2007). Context sensitive stemming for web search // Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 639–646.
9. Majumder P., Mitra M., Swapan K., Kole P G., Mitra P., Datta K (2007). “YASS: Yet another suffix stripper” // ACM Transactions on Information Systems. 25 (4) 18.
10. Adam G., Asimakis K., Bouras C., Poulopoulos V (2010). An efficient mechanism for stemming and tagging: the case of Greek language // In the Proc. of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems: Part III, pp 389-397.
11. Feinerer I (2010). Analysis and Algorithms for Stemming Inversion. In: Cheng PJ., Kan MY., Lam W., Nakov P. (eds) // Information Retrieval Technology. AIRS 2010 // Lecture Notes in Computer Science vol. 6458. Springer, Berlin, Heidelberg.
12. Jiaul H. P., Mitra M., Swapan K. P., Järvelin K (2011). GRAS: An effective and efficient stemming algorithm for information retrieval // ACM Transactions on Information Systems, 29 (4), 1-24.
13. Fernández A., Díaz J., Gutiérrez Y., Muñoz R (2011). An Unsupervised Method to Improve Spanish Stemmer. In: Muñoz R., Montoyo A., Métais E. (eds) // Natural Language Processing and Information Systems // Lecture Notes in Computer Science. Vol. 6716. Springer, Berlin, Heidelberg.
14. Madani A., M. Kissi M (2014). Building a syntactic rules-based stemmer to improve search effectiveness for Arabic language // 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14), pp. 1-6.
15. Danilova V., Alexandrov M., Blanco X (2014). A Survey of Multilingual Event Extraction from Text. // In: Métais E., Roche M., Teisseire M. (eds) Natural Language Processing and Information Systems // Lecture Notes in Computer Science, Vol. 8455. Springer, Cham.
16. Moral C., de Antonio A., Imbert R., Ramírez J (2014). A survey of stemming algorithms in information retrieval // Information Research 19(1), 605.
17. Loukachevitch, N V., Chetviorkin, I (2014). Open evaluation of sentiment-analysis systems based on the material of the Russian language // Scientific and Technical Information Processing, 41(6), 370-76.
18. Gadri S., A Moussaoui A. (2015). Information retrieval: A new multilingual stemmer based on a statistical approach // 3rd International Conference on Control, Engineering & Information Technology, Tlemcen, Algeria, pp.1-6.
19. Brychcín T., Konopík M. (2015). HPS: High precision stemmer // Information Processing and Management, 51 (1), 68-91.
20. Singh J., Gupta V (2016). Text Stemming: Approaches, Applications, and Challenges // ACM Computing Surveys (CSUR), 49 (3), Article 45.
21. Beltiukov A.P., Abbasi M.M (2019). Logical analysis of emotions in text from natural language // Vestnik Udmurtskogo Universiteta Matematika Mekhanika Komp'yuternye Nauki, 29 (1), 106-116.
22. Bölücü N., Burcu C (2019). Unsupervised Joint POS Tagging and Stemming for Agglutinative Languages // ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18 (3), Article 25.
23. Porter M.F (2001). Snowball: A language for stemming algorithms // Published online, (October 2001) Accessed 8.11.2021, 15.00h. http://snowball.tartarus.org/texts/introduction.html
24. *Лукашевич Н. В., Левчик А. В*. Создание лексикона оценочных слов русского языка РуСентилекс // Труды конференции OSTIS-2016. 2016. С. 377–382.
25. Loukachevitch N., Levchik A (2016). Creating a General Russian Sentiment Lexicon. // In the Proc. of Language Resources and Evaluation Conference LREC-2016.
26. Список чувств и эмоций : блог психолога Петра Зарубина из г. Новосибирска. URL: https://peterzarubin.ru/spisok-chuvstv-i-emotsij.

References

1. Rijsbergen J., Robertson C. J., Stephen E., (1946) & Porter, Martin F. (1980). New models in probabilistic information retrieval // British Library Research and Development Dept., [London]. No. 5587.
2. Porter M.F. An algorithm for suffix stripping (1980). Emerald Publishing, Program 1 14 (3), 130-137.
3. Krovetz R (2000). Viewing morphology as an inference process // Artificial Intelligence Journal, Q1 SJR 1.01. 118(1), 277-294.
4. Paice C. D (1990). Another Stemmer // ACM SIGIR Forum, 24(3), 56-61.
5. William B. Frakes , Christopher J (2003). Fox. Strength and similarity of affix removal stemming algorithms // ACM SIGIR Forum, 37(1), 26-30.
6. Bacchin M., Ferro N., Lucci M (2005). A probabilistic model for stemmer generation // Information Processing and Management 41(1), 121-137.
7. Wiebe, J., Wilson T., Cardie C (2005). Annotating expressions of opinions and emotions in language // Language Resources and Evaluation 39 (2), 165-210.
8. Peng, F., Ahmed, N., Li, X., Lu, Y (2007). Context sensitive stemming for web search // Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 639–646.
9. Majumder P., Mitra M., Swapan K., Kole P G., Mitra P., Datta K (2007). "YASS: Yet another suffix stripper" // ACM Transactions on Information Systems. 25 (4) 18.
10. Adam G., Asimakis K., Bouras C., Poulopoulos V (2010). An efficient mechanism for stemming and tagging: the case of Greek language // In the Proc. of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems: Part III, pp 389-397.
11. Feinerer I (2010). Analysis and Algorithms for Stemming Inversion. In: Cheng PJ., Kan MY., Lam W., Nakov P. (eds) // Information Retrieval Technology. AIRS 2010 // Lecture Notes in Computer Science vol. 6458. Springer, Berlin, Heidelberg.
12. Jiaul H. P., Mitra M., Swapan K. P., Järvelin K (2011). GRAS: An effective and efficient stemming algorithm for information retrieval // ACM Transactions on Information Systems, 29 (4), 1-24.
13. Fernández A., Díaz J., Gutiérrez Y., Muñoz R (2011). An Unsupervised Method to Improve Spanish Stemmer. In: Muñoz R., Montoyo A., Métais E. (eds) // Natural Language Processing and Information Systems // Lecture Notes in Computer Science. Vol. 6716. Springer, Berlin, Heidelberg.
14. Madani A., M. Kissi M (2014). Building a syntactic rules-based stemmer to improve search effectiveness for Arabic language // 9Proc. of the 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14), pp. 1-6.
15. Danilova V., Alexandrov M., Blanco X (2014). A Survey of Multilingual Event Extraction from Text. // In: Métais E., Roche M., Teisseire M. (eds) Natural Language Processing and Information Systems // Lecture Notes in Computer Science, Vol. 8455. Springer, Cham.
16. Moral C., de Antonio A., Imbert R., Ramírez J (2014). A survey of stemming algorithms in information retrieval // Information Research 19(1), 605.
17. Loukachevitch, N V., Chetviorkin, I (2014). Open evaluation of sentiment-analysis systems based on the material of the Russian language // Scientific and Technical Information Processing, 41(6), 370-76.
18. Gadri S., A Moussaoui A. (2015). Information retrieval: A new multilingual stemmer based on a statistical approach // 3rd International Conference on Control, Engineering & Information Technology, Tlemcen, Algeria, pp.1-6.
19. Brychcín T., Konopík M. (2015). HPS: High precision stemmer // Information Processing and Management, 51 (1), 68-91.
20. Singh J., Gupta V (2016). Text Stemming: Approaches, Applications, and Challenges // ACM Computing Surveys (CSUR), 49 (3), Article 45.
21. Beltiukov A.P., Abbasi M.M (2019). Logical analysis of emotions in text from natural language // Vestnik Udmurtskogo Universiteta Matematika Mekhanika Komp'yuternye Nauki, 29 (1), 106-116.
22. Bölücü N., Burcu C (2019). Unsupervised Joint POS Tagging and Stemming for Agglutinative Languages // ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18 (3), Article 25.
23. Porter M.F (2001). Snowball: A language for stemming algorithms // Published online, (October 2001) Accessed 8.11.2021, 15.00h. http://snowball.tartarus.org/texts/introduction.html
24. Loukachevitch N., Levchik A [Lexicon creation for the evaluation of words of Russian language RuSentilex]. *Trudy konferentsii OSTIS-2016* [Proc. of the conference OSTIS-2016]. 2016. Pp. 377-382 (in Russ.).
25. Loukachevitch N., Levchik A (2016). Creating a General Russian Sentiment Lexicon. // In the Proc. Of Language Resources and Evaluation Conference LREC-2016.
26. Blog article by psychologist Peter Zarubin from the city of Novosibirsk, title «Feelings and Emotions». Available at: https://peter-zarubin.ru/spisok-chuvstv-i-emotsij (accessed 05.10.2021, 10.00h.).

* * *

Emostemmer: An Effective Program for Determining Emotions in Russian Using N-grams (Emotiograms)

M. M. Abbasi, Post-graduate, Udmurt State University, Izhevsk, Russia
A. P. Beltyukov, DSc (Physics and Mathematics), Professor, Udmurt State University, Izhevsk, Russia

Emotions and the analysis of their expression in texts is a topic of growing interest in recent years. Researchers are trying to create an intelligent machine that can not only read the text, but also determine its emotional state. The results obtained can be used to prepare the machine for future predictions of the emotional orientation of texts, their authors and readers. This text analysis can also be used to get feedback from people about a product or service, reaction to an event or government policy, etc. It includes syntactic as well as semantic text analysis. Parsing consists of identifying words that represent emotions in a text. For its identification, the stemmer plays an important role - the stem or root of the word. In many languages of the Romano-Germanic group, the identification of words representing emotions is much easier than in Russian, since one word represents emotion regardless of grammatical forms and genders. While for a language such as Russian, where the ending of an emotionally charged word changes depending on the genus, species, etc., the analysis becomes more complex. There are different methods of defining emotions in a text.

This work focuses on identifying emotions from the text while limiting the complexity of the algorithm by requiring a minimum amount of memory and time. We have created the Emostemmer program, which is an N-gram stemmer (in which letters from words are grouped in a sequence of 2 letters, 3 letters… ..N letters called N-grams) to identify words that represent emotions in the text. The performance of Emostemmer versus RuSentiLex was determined by training and testing a support vector machine classifier with both algorithms. The results of the work are described in detail below in the "Methodology" and "Discussion" sections.

Keywords: text, emotions, blog, communication, stemming, analysis, confusion matrix.

Получено: 01.12.21