Санкт–Петербургский государственный университет

БУТЕНКО Александр Алексеевич

Выпускная квалификационная работа

Методы детектирования и визуализации информации о геолокации пользователей в дискуссиях в социальных сетях

Научный руководитель:
кандидат технических наук,
кафедра технологии программирования
Блеканов Иван Станиславович

Рецензент:
кандидат физико-математических наук,
кафедра компьютерных технологий и систем
Коровкин Максим Васильевич

Санкт-Петербург
2021 г.

Содержание

Введение

Актуальность

Сервисы социальных сетей в интернете, такие как микроблоги, предлагаемые такими платформами, как Twitter, продемонстрировали феноменальный рост своей пользовательской базы. Этот рост вызвал интерес к использованию данных, предоставляемых этими платформами, для извлечения разного рода информации, такой как, например, географическое положение, от пользователей. Полученные данные можно использовать для предоставления пользователям персонализированных услуг, таких как релевантные новости, реклама и прочий контент. Также знания о местоположении пользователей могут позволить исследователям анализировать мировые события с точки зрения того, как и какие слои населения они задевают. С более чем 200 миллионами учетных записей в Twitter в разных географических точках короткие сообщения (твиты) образуют огромный набор данных, который может быть проанализированы для извлечения такой географической информации.

Проблемы

Twitter позволяет своим пользователям самостоятельно указывать свое географическое положение. Эта информация о местоположении вводится пользователем вручную или обновляется с помощью GPS (но активировано это лишь у небольшого процента пользователей [1]). Следовательно, данные о географическом местоположении для большинства пользователей могут отсутствовать или быть неверными. Есть несколько недостатков в использовании обновления местоположения вручную:

- основной и единственный инструмент для получения информации о геолокации пользователей - официальный сервис Twitter API. К сожалению, доступ к данному сервису открыт исключительно на платной основе.

- пользователи могут ввести неверные данные о географическом местоположении. Например, пользователь может ввести свое местоположение как “Марс, кратер Ковальский”. Также это может быть не название реального географического местоположения “Криптон”;

- пользователи могут неоднозначно указать свое местоположение. Например, “космополит, но из Москвы”. Такую строку сложно обработать, так как в ней много лишней информации;

- у пользователей может быть не указано местоположение.

Следовательно, надежность таких данных для определения географического местоположения пользователя невысока. Чтобы преодолеть эту проблему редко доступной информации о местоположении пользователей, мы оцениваем географическое положение пользователя Twitter на уровне страны, основываясь не только содержании указанного поля местоположения, но и на геолокации пользователей, с которыми взаимодействует рассматриваемый.

Цель и задача работы

В Twitter пользователи могут публиковать микроблоги, известные как твиты, которые могут читать другие пользователи. Наряду с этой службой микроблогов Twitter также предоставляет службу социальной сети, в которой пользователь (подписчик) может следить за твитами другим пользователем. Каждый край социальной сети формируется этими отношениями «подписки». Как подписчик, пользователь получает все твиты, отправленные подписчиком, и, в свою очередь, может отвечать на эти твиты с помощью ответного твита. Этот ответ-твит - ключевой инструмент взаимодействия пользователей Twitter, который составляет основу разговора между двумя разными пользователями. Согласно исследованию [2] ответные твиты и направленные твиты составляют около 25,4% всех сообщений в Twitter. Это показывает, что функция ответа-твита широко используется пользователями Twitter. Основная идея моей работы заключается в том, что так как разговор между пользователями может быть посвящен темам

(погода, спорт и т. д.) связанным с местоположением, предполагается, что этот набор тем остается неизменным во время обсуждения. Тогда можно предполагать, что геолокация участника дискуссии связана с геолокацией остальных участников. Таким образом, составив граф того, с кем пользователь взаимодействует в Twitter, можно составить представление о его местоположении.

Подытоживая, методологию научной работы можно разделить на 3 этапа:

- обработка верно указанных геопозиций;
- обработка геопозиций, содержащих грязные данные;
- обработка пользователей при помощи графовых вероятностных методов.

Цель данной работы заключается в решении описанных выше проблем посредством разработки программного комплекса, который позволял бы получать необходимые геоданные от пользователей без необходимости получать доступ к Twitter API. Также сервис должен иметь возможность обрабатывать как грязные данные от пользователей, так и определять геолокацию пользователей без указанной в явном виде геолокации.

Задачу данной работы можно формализовать, описав тремя основными этапами:

- Проектирование архитектуры программного обеспечения;
- Реализация необходимых алгоритмов;
- Оценка качества полученного ПО.

Обзор литературы

В силу того, что работа состоит из разных этапов, для ее реализации были изучены различные источники разной направленности.Изученная система Bloom Embedding представлена в данной статье [3] . Residual Neural Network рассмотрена в оригинальной статье авторов данной архитектуры [4] . Процесс обучения мультиязычной системы для распознования именных сущностей, реализованной в библиотеке spaCy, описывается в следующем материале [5] . Архитиктура word2vec, сравниваемая в данной работе с Bloom Embedding, представлена в статье [6] . В процессе исследования области векторизации слов также были исследованы такие архитектуры как fasttext [7] и GloVe [8] . Для нормализации строк геолокаций использовался Open Street Maps API, описанный в следующей документации [9] .

Для ознакомления с текущими разработками в смежной области был изучен ряд статей. [10] - обзорная статья, посвященная social network analysis (SNA). В статье [2] описывается подход, в котором геолокация пользователя определяется исходя из содержания его твитов. Здесь [11] авторы статьи предсказывают партию, к которой принадлежит член сената США, основываясь на его взаимодействиях в Twitter. В данной статье [12] авторы исследовали динамику мнений людей в зависимости от их взаимодействий с другими мнениями. [13] исследует влияние событий в мире на взаимодействия в социальных сетях на примере хэштега #BoycottNFL. [14] посвящена рекомендациям товаров для пользователей в соответствии с их поведением на сайте магазине.

Глава 1. Разработка программного комплекса для определения геолокации пользователей в дискуссиях в в социальных сетях

Проектирование архитектуры программного обеспечения

Для реализации поставленных задач была выбрана микросервисная архитектура, позволяющая представить каждый этам обработки данных как отдельный сервис. Такой подход обеспечивает модульность системы, что позволяет эффективнее распределять задачи и структурировать работу над проектом.

Алгоритм геолокации пользователей

Алгоритм обработки дискуссий в данной работе состоит из нескольких этапов:

- получение указанной пользователем информации о геолокации в соответствии с его уникальным именем;
- обработка полученной информации при помощи бейзлайн решения (в нашем случае - Open Street Maps API);
- улучшение результатов предыдущего шага при помощи применения методов распознования именованных сущностей;
- определение оставшихся геолокаций при помощи графа связей пользователей в дискуссии.

В следующих секциях подробнее рассматривается каждый из этих этапов.

Получение геолокации по имени пользователя

Для получения информации о геолокации, соответствующей конкретному пользователю, генерируется запрос, имитирующий тот, который отсылается при заходе на страницу пользователя. Ответ возвращается в ви-

де JSON-файла, содержащего всю информацию со страницы пользователя, что позволяет в результате обработки получить значения поля с локацией.

Во время обработки списка участников дискуссии мы столкнулись с тем, что многие из указанных имен пользователей либо уже не зарегестрированы в базе данных Twitter, либо пользователи заблокированы. При обработке мы отсылаем запросы к API с целью получить информацию об имени пользователя. В случае описанных выше имен API вернет нам ошибку '50' в случае, когда пользователя с таким username нет в системе, и ошибку '63' , если пользователь заблокирован. Обработать поле геолокации у таких пользователей у нас не получится, поэтому на следующие 2 этапа мы просто их не рассматриваем.

Обработка верно указанных геопозиций

Обработка верно указанных заключалась в приведении их к стандартному виду ISO 3. Для это было использовано API некоммерческого веб-картографического проекта Open Street Maps (далее OSM), а именно Nominatim - инструмент для поиска данных OSM по имени и адресу географического места. Nominatim принимает на вход строку с названием объекта и в случае успешной обработки возвращает JSON, содержащий двухбуквенный код страны, в которой находится объект. Этот код в дальнейшем может быть преобразован в ISO 3 при помощи словаря зависимостей. Важно отметить, что каждая обработанная строка сохраняется в базу данных, так что если она встретится в поле местоположения другого пользователя алгоритм сразу вернет нам ее в обработанном виде, предотвращая лишние запросы к API.

Обработка геопозиций, содержащих грязные данные

Сейчас наш алгоритм способен обработать только чистые данные . В реальной жизни пользователи чаще вводят свое местоположение самостоятельно, что приводит к тому, что эти данные невозможно обработать при помощи OSM Nominatim. Рассмотрим на примере строки "Я из Ростова-на-Дону". Обработка данной строки через API не дает никаких результатов,

однако обработка строки "Ростова-на-Дону"возвращает нам верный ответ.

```
[163]: nominatimQueryToCountryCode('Ростова-на-Дону')

[163]: 'ru'
```

Получаем, что если бы у нас была возможность выделить лишь необходимую информацию из строки, мы могли бы и дальше использовать OSM API.

Тут нам на помощь приходит система по распознаванию именованных сущностей (Named Entity Recognition, далее NER), реализованная в библиотеке spaCy для языка python. NER позволяет не только определить именованные сущности, но и отнести их к одному из лейблов:

- LOC (Location) - название локации (город, страна, область и т.д.);
- ORG (Organisation) - название организации;
- PER (Person) - имя, фамилия и т.д.;
- MISC (Miscellaneous entities) - прочие различные наименования (праздники, национальности, продукты, произведения искусства и т.д.).

Pipeline обработки текста в случае spaCy - Transition-based NER . Данная архитектура описана в статье [15]. Идея заключается в следующем. У нас есть буфер слов в предложении и массив слов состояния (изначально пустой). Также у нас определены операции над состояниями:

- shift - добавить следующее слово в массив состояния;
- reduce - выводит текущие слова в массиве состояния с определенной меткой и очищает массив состояния;
- out - выводит следующее слово из буфера, не помещая его в массив состояния.

Операция	Вывод	Состояние	Буфер	Предсказание
	[]	[]	[Джет, Ли, в, Москве]	
SHIFT	[]	[Джет]	[Ли, в, Москве]	
SHIFT	[]	[Джет, Ли]	[в, Москве]	
REDUCE(PER)	[PER(Джет Ли)]	[]	[в, Москве]	PER
OUT	[PER(Джет Ли), в]	[]	[Москве]	
SHIFT	[PER(Джет Ли), в]	[Москве]	[]	
REDUCE(LOC)	[PER(Джет Ли), в, LOC(Москве)]	[]	[]	LOC

В этих условиях задача NER заключается в предсказании следующей операции. Matthew Honnibal, создатель компании Explosion AI, создателей spaCy, выделяет тут четыре этапа в реализации алгоритма распознования именованных сущностей:

- Эмбеддинг (embedding) - векторизация слов документа;
- Энкодинг (encoding) - приведение векторов, полученных на первом этапе, к матрице, где каждая строка представляет слово в контексте рассматриваемого предложения;
- Векторизация (attend) - приведение полученной на втором этапе матрицы предложения к виду вектора для дольнейшей обработки;
- Предсказание - классификация слов в предложении по полученному вектору предложения.

Пройдемся по каждому из этапов. На первом этапе мы переводим каждое из слов в векторное пространство. Основная интуиция обучения модели, переводящей слова в векторы, заключается в предположении, что векторы близких по значению слов должны находится близко в векторном пространстве. В соответствии с этой логикой, были реализованы многие модели эмбеддинга слов. Одна из классических моделей - это word2vec. Word2vec обучается следующим образом:

- Изначально каждому уникальному слову в корпусе присваивается вектор длины n, где n - количество уникальных слов в корпусе. Все значения этого вектора равны нулю, кроме равного единице значения под номером i, соответствующего номеру кодированного слова. Данный подход называется one-hot encoding.

- Далее мы строим линейную модель из двух матриц: размерности $[n \times k]$ и размерности $[k \times n]$. В данной модели k - размер желаемого вектора для слова.

- Полученная модель затем обучается на корпусе текста. Вектор каждого слова из текста используется в качестве входного вектора модели, а в качестве истинных значений используются вектора l слов справа и l слов слева, где l - произвольное значение. Таким образом для каждого слова из корпуса мы производим $2l$ обучающих примеров.

- В результате обучения мы получаем матрицу $[n \times k]$, где i-ая строка представляет собой вектор для i-ого слова.

Данный подход имеет 2 существенных недостатка:

- Размерности матриц напрямую зависят от количества уникальных слов в корпусе, соответственно для больших корпусов текста использовать такой подход вычислительно неэффективно.

- Модель способна векторизовать только те слова, которые встречаются в обучающем корпусе. Более того, даже если в корпусе и встречалось слово, но в другой форме, вектор для него получить не получится.

Обе описанные проблемы решаются при помощи алгоритма Bloom Embedding. Очевидно, что обе описанные выше проблемы связаны с размерностями матрицы эмбеддингов. Идея Bloom Embedding заключается в использовании матрицы эмбеддингов фиксированной размерности. Реализовано это следующим образом:

- определяем хэш-функций H_i, $i = \overline{1, k}$;

- каждая хэш функция принимает на вход слово и возвращает нам значение u_i, $u = \overline{1, l}$. Здесь l - желаемое количество строк в матрице ембеддингов;

- из матрицы ембеддингов выбираются строки, соответствующие номерам u_i, после чего конкатенируются, образуя эмбеддинг для слова. Важно отметить, что полученная в результате обработки система способна векторизовать даже те слова, которых не было в обучающей выборке.

Следующий этап обработки строки - энкодинг. Нам необходимо выразить слова через векторы, которые учитывают их контекст. Для этой задачи в библиотеке spaCy используется нейронная сеть с архитектурой ResNet (Residual Neural Network). Алгоритм:

- сеть принимает на вход три конкатенированных эмбеддинга слов - кодируемое и соседние с обеих сторон;
- конкатенированные векторы пропускаются через полносвязную нейронную сеть;
- в результате выполнения слоя возвращается сумма входа и выхода слоя;
- алгоритм повторяется заданное количество раз (в случае spaCy - 4 раза).

На каждом слое данной сети рецептивное поле увеличивается на 1 с двух сторон. Так, в результате обработки слова в нашем случае новый вектор слова учитывая контекст из 4 слов с каждой из сторон.

На следующем этапе обработки векторизуем наше текущее состояние. Для это из буфера мы береме вектора передыдущего, текущего и следующего слов, а также вектора двух последних распознанных именованных сущностей. Для отобранных векторов применяем функцию maxout, в результате чего получаем вектор состояния.

На последнем этапе полученный на этапе векторизации вектор пропускается через полносвязную нейронную сеть, в результате чего предсказывается возможное действие над состоянием.

В результате описанной выше обработки мы получим исходный текст с метками для распознанных именных сущностей. В нашей задаче нас интересует исключительно лейбл LOC. Мы применяем к полю местоположения NER и в случае нахождения части предложения с лейблом локации пропускаем ее через Nominatim.

Обработка данных, не содержащих полезной информации

В результате обработки, описанной выше, у нас остались необработанными два вида полей:

- поля, не содержащие информации о местоположении пользоваетеля
- пустые поля

Тут мы можем воспользоваться данными о взаимосвязи пользователей в дискусси. Будем присваивать пользователям те геолокации, с участниками дискусси из которых они взаимодействуют больше всего. Интуитивно, даже если это не позволит нам непосредственно определить геолокацию пользователя, мы все равно сможем определить, какую из сторон дискусси он представляет. В идеале, в результате мы получим геолокацию для каждого пользователя из списка вершин. Однако, важно отметить, что некоторые пользователи все еще могут быть неопознаны (в том случае, когда вершина либо изолирована, либо соединена исключительно с неопознаными вершинами, не имеющими пути к опознанным вершинам). Данный подход оценивался следующим алгоритмом:

- сначала алгоритм запускается на наборе данных со всеми определенными на предыдущих этапах геолокациями;
- затем алгоритм запускается на наборе данных, содержащем 80% определенных на предыдущих этапах данных;
- результаты сравниваются в точках, определенных на ранних этапах, при помощи метрики точности и f1.

Средние результаты по пяти оценкам:

- точность: 0,96
- f1: 0,86

Формат данных

Данные для эксперимента представлены в виде ориентированного графа, где пользователи представляют собой вершины графа, а взаимосвязи пользователей - ребра графа. Таким образом, для каждого рассматриваемого события (дискуссии) у нас представлено два файла:

- Nodes.csv - набор вершин (уникальные имена участников дискуссии)
- Edges.csv - набор направленных ребер (взаимосвязи этих пользователей)

При помощи уникальных имен мы можем определить локации тех пользователей, которые заполнили данную информацию в своем профиле, а благодаря информации о взаимосвязи пользователей у нас есть возможность детектировать геолокацию для тех вершин, в которых она не указана в явном виде.

Оценка результатов

В результате проведенного исследования были обработаны более 50 000 пользователей социальной сети Twitter: более 40 000 пользователей в дискуссии посвященной беспорядкам в Германии и более 10 000 пользователей в дискуссии посвященной беспорядкам в Бирюлево. Для каждого этапа обработки были собраны статистики, отражающие количество успешно обработанных пользователей и их долю в общем количестве пользователей. Собранная статистика отражена в таблице ниже.

	Бирюлево		Кельн	
	Количество	Доля	Количество	Доля
Результат обработки OSM API без NER	4 117	0.36	11 868	0.30
Результат обработки OSM API с применением NER	4 573	0.4	14 799	0.37
Результат заполнения графа в соответствии с заполненными нодами	10 631	0.93	37 805	0.94
Всего	11 429		40 117	

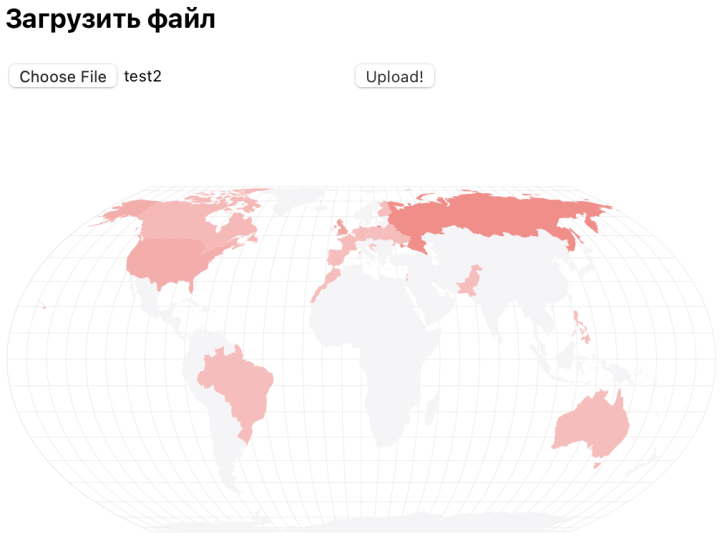
Нетрудно заметить, что предложенный метод обработки при помощи распознования именованных сущностей добавляет 10-20 % от опознанных без нормализации полей геолокации данных. Графовый метод в свою очередь позволяет предсказывать более чем половину геолокаций пользователей в дискуссии, основываясь лишь на 40 % опознанных.

Веб-сервис

Для визуализации результатов было реализовано веб-приложение, позволяющее загрузить список уникальных имен участников дискуссии и на выходе получить отображенную на карте задействованность пользователей из разных стран.

При загрузке сайт выглядит так:

Пользователю предлагается выбрать файл с username'ами. После этого можно загрузить файл. По результатам обработки на карте отображаются страны, встретившиеся в нашем наборе. Чем темнее, тем больше людей в дискуссии из этой страны.



Для реализации front-end части программного обеспечения использовался язык JavaScript вместе с библиотекой React JS. Back-end реализован на языке python при помощи библиотеки Flask.

Заключение

Результаты

В данной работе рассматривалась задача детекции геолокации пользователей в социальной сети Twitter. В изучаемых данных были представлены графы дискуссий, посвященные различным нашумевшим мировым событиям. Разработанный в результате исследования алгоритм способен определить геолокацию для пользователей, которые указали данную информацию о себе, а также предсказать геолокацию для пользователей без подобной информации.

В ходе работы были выполнены все стадии обработки рассматриваемых данных. Был реализован сбор информации о поле геолокации пользователей посредством скрапинга Twitter API. Также было произведено распознование именованных сущностей из полученных через Twitter API строк местоположений. Затем была реализована нормализация распознанных геолокаций при помощи Open Street Maps API. Последним шагом в обработке данных было заполнение графа связей пользователей геолокациями в соответствии с определенными на предыдущих шагах местоположениях. Были собраны результаты работы алгоритма на разных датасетах, отражающие результативность алгоритма на каждом этапе обработки. Также был реализован web-сервис, в функционал которого входит:

- загрузка данных об участниках дискуссии,
- обработка этих данных при помощи реализованного алгоритма,
- визуализация результатов обработки алгоритма.

Полученная система может быть применена для анализа вовлеченности пользователей из разных стран в обсуждения мировых событий.

Перспективы развития

Представленная работа имеет следующие потенциальные пути развития:

- Обработка информации о лайках и ретвитах. Использование этой информации может позволить как расширить множество участников дискуссии, так и увеличить точность предсказаний.
- Обработка информации о подписках пользователя. Данный подход также может увеличить точность предсказываний.
- Использование семантических свойств самих сообщений в дискуссии.
- Использование гео-меток сообщений пользователей.

Список литературы

[1] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, (New York, NY, USA), p. 759–768, Association for Computing Machinery, 2010.

[2] S. Chandra, L. Khan, and F. B. Muhaya, "Estimating twitter user location using social interactions–a content based approach," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 838–843, 2011.

[3] J. Serrà and A. Karatzoglou, "Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, (New York, NY, USA), p. 279–287, Association for Computing Machinery, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[5] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, "Learning multilingual named entity recognition from wikipedia," *Artificial Intelligence*, vol. 194, pp. 151–175, 2013. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2017.

[8] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[9] O. Wiki, "Research — openstreetmap wiki,," 2021. [Online; accessed 20-May-2021].

[10] D. Camacho, Ángel Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, and E. Cambria, "The four dimensions of social network analysis: An overview of research methods, applications, and software tools," *Information Fusion*, vol. 63, pp. 88–120, 2020.

[11] J. M. Chamberlain, F. Spezzano, J. J. Kettler, and B. Dit, "A network analysis of twitter interactions by members of the u.s. congress," *Trans. Soc. Comput.*, vol. 4, Feb. 2021.

[12] T. Tang and C. G. Chorus, "Learning opinions by observing actions: Simulation of opinion dynamics using an action-opinion inference model," *Journal of Artificial Societies and Social Simulation*, vol. 22, no. 3, p. 2, 2019.

[13] T.-L. D. Chung, O. Johnson, A. Hall-Phillips, and K. Kim, "The effects of offline events on online connective actions: An examination of boycottnfl using social network analysis," *Computers in Human Behavior*, vol. 115, p. 106623, 2021.

[14] L. Ren, B. Zhu, and Z. Xu, "Robust consumer preference analysis with a social network," *Information Sciences*, vol. 566, pp. 379–400, 2021.

[15] X. Dai, S. Karimi, B. Hachey, and C. Paris, "An effective transition-based model for discontinuous ner," 2020.