Санкт-Петербургский государственный университет
Факультет Прикладной математики - Процессов управления

Кафедра Технологии программирования

Пекша Любовь Станиславовна

Выпускная квалификационная работа бакалавра

Название

Направление 01.03.02
Прикладная математика и информатика

Научный руководитель:
доцент Блеканов И.С.

Санкт-Петербург
2019

Оглавление

Введение **3**

Постановка задачи **4**

Обзор литературы **5**

1. Данные **7**

1.1. Сбор данных 7

1.2. Формирование траекторий 9

1.3. Обзор данных 10

2. Базовая модель **11**

2.1. Описание модели 11

2.2. Достоинства и недостатки 14

3. Модернизация базовой модели **15**

3.1. Изменение рекуррентного слоя 15

3.1.1. Идея 15

3.1.2. Результаты 15

3.2. Изменение функции ошибки 17

3.2.1. Идея 17

3.2.2. Результаты 18

3.3. Дополнительные изменения 19

4. Сравнение моделей **20**

Выводы **21**

Заключение **22**

Список литературы **23**

Введение

Digital Humanities (цифровые гуманитарные науки) — стремительно развивающееся в наше время направление. На это есть множество причин, например успехи в области анализа данных, появление открытых данных, практическая польза, а также желание исследователей применить имеющийся математический аппарат в новой сфере.

Анализ текстов СМИ и анализ упоминаний персоналий в СМИ в частности — слабо затрагиваемая область Digital Humanities (цифровые гуманитарные науки), которая не только представляет собой не только интересную с научной точки зрения задачу, но и имеет практическое применение. Например, сравнение упоминаний персоналии в различных СМИ или получение общего представления о личности путем анализа СМИ. Также особый интерес представляет собой анализ персоналии во времени, то есть изучение того, как изменяется новостная повестка относительно персоналии с течением времени.

Причины по которым данная сфера не исследована кроются в следующем: отсутствие размеченных данных, отсутствие способа формально измерить качество работы алгоритма, сравнительно небольшое количество работ посвященных анализу текста во времени в целом

Настоящая работа посвящена анализу упоминаний персоналий в СМИ в долгосрочном периоде. Данная тема предоставляет широкий спектр возможных направлений для работы такие как выделение траекторий персоналий — ранжированного по времени набора статей о персоналии, выявление долгосрочных трендов траектории, а также анализ эмоциональной окраски траекторий.

Постановка задачи

Основной задачей проекта является реализация существующей модели осуществляющей тематико-эмоциональный анализ траекторий персоналий в долгосрочном периоде и ее применение для публицистических текстов, ее модернизация и анализ полученных результатов.

Для достижение цели ставятся следующие задачи:

1. Формирование обучающей выборки
2. Построение базовой модели, способной выделять долгосрочные тренды и тематики
3. Модернизация модели
4. Анализ полученных результатов

Обзор литературы

К сожалению, на данный момент отсутствуют подходы, реализующие анализ траекторий СМИ.

Схожие модели реализованы для художественной литературы[1, 2].

Особый интерес представляет собой работа Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships[1]. В этой работе представляется модель **RMN**, которая совместно обучает набор глобальных дескрипторов отношений между героями из художественных текстов, а также сопоставляет каждому отрывку текста веса дескрипторов, которые отражают насколько хорошо каждое из слов-дескрипторов описывает траекторию в текущий момент времени.

RMN превзошла модель **HTMM**[2], решающую сходную задачу, а также модели тематического моделирования **LDA**[3] и **NUBBI**[4], которые способны выделять тематики в неразмеченном тексте, но не предназначены для построения траекторий в долгосрочном периоде.

Данная модель решает следующие подзадачи:

1. Выделение интерпретируемых дескрипторов — тематического базиса для составления траекторий

2. Создание модели, способной сопоставлять траектории распределение на дескрипторах (веса дескрипторов), которое отражает долгосрочные тренды в текстах статей траектории а также их изменение

RMN хорошо справляется со своей задачей, выделяя интерпретируемые дескрипторы, среди которых достаточно четко можно выделить эмоциональные (love, sadness, violence, etc.).

Несмотря на свои достоинства, архитектура данной модели имеет ряд недостатков, которые будут рассмотрены далее.

Возможно что-то стянуть из описания базовой модели

1. Данные

Для формирование обучающей выборки необходимо выполнить следующие шаги

1. Сбор данных с сайтов СМИ
2. Выделение траекторий персоналий - ранжированного по времени набора статей о персоналии
3. Обработка текста статей с целью выделить релевантную информацию

Для дальнейшей работы необходимо собрать данные с сайтов СМИ. В открытом доступе есть датасеты со статьями различных СМИ. Но в силу того, что такие датасеты не предоставляют информацию о дате выхода статьи, возникла необходимость собрать необходимые данные непосредственно с новостных сайтов.

Для работы были выбраны следующие источники: Lenta.ru[8], Tvrain[9], Meduza[10], РИА Новости[11]. Данные СМИ имеют внушительный архив данных, а также ведут свою деятельность продолжительный период времени.

1.1. Сбор данных

Сбор данных осуществлялся с помощью Scrapy[12]. Это написанная на Python платформа, которая нацелена на простой, быстрый и автоматизированный обход (краулинг) веб-страниц, имеющий большую популярность. Одним из главных преимуществ Scrapy является то, что он построен поверх Twisted, асинхронного сетевого фреймворка. Ассинхронность означает, что не нужно ждать завершения запроса, прежде чем сделать еще один, это позволяет добиться высокого уровня производительности. Тот факт, что Scrapy реализован

с использованием неблокирующего (асинхронного) кода для параллелизма, делает его одним из самых эффективных фреймворков для краулинга.

В ходе работы по сбору данных возникло несколько проблем.

Первая из них — отсутствие архива статей на сайте СМИ Meduza[10]. Данная проблема была решена с помощью группы ВКонтакте данного интернет-издания. Ссылки на новостные статьи были собраны при помощи VK API — интерфейса, который позволяет получать информацию из базы данных vk.com. Для удобства работы с VK API была использована библиотека vk для Python. Эта библиотека предоставляет удобный интерфейс для работы с VK API, а также не требует авторизации.

Еще одна проблема возникла из-за того, что сайт телеканала Дождь блокирует запросы от краулера (программы, осуществляющей сбор данных), если эти запросы поступают от него слишком часто. Из-за этого краулер без дополнительных изменений будет скачивать лишь часть доступной информации. Выход из этой ситуации – использовать разные useragent для сетевых запросов. Useragent — это клиентское приложение, использующее определённый сетевой протокол. При посещении веб-сайта клиентское приложение обычно посылает веб-серверу информацию о себе. Это текстовая строка, являющаяся частью HTTP запроса, обычно включающая такую информацию, как название и версию приложения, операционную систему компьютера и язык. Библиотека fake_useragent для Python позволяет создавать случайные useragent для каждого запроса на сайт.

Для хранения полученных данных используется база данных на основе SQLite. Для работы с базой данных используется SQLAlchemy. Это набор инструментов SQL с открытым исходным кодом и ORM (технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования)

для языка программирования Python.

1.2. Формирование траекторий

Для формирования траекторий необходимо уметь извлекать из текста имена упоминаемых в нем людей. Для этого необходимо использовать программу, решающую задачу извлечения именованных сущностей. Извлечение именованных сущностей — это класс подзадач извлечения информации, цель которой найти и классифицировать упоминания именованных сущностей в неструктурированном тексте по заранее определенным категориям, таким как имена людей, организации, адреса, даты и т. д. Библиотека DeepPavlov предоставляет модель, которая решает задачу извлечения имен для текстов на русском языке.

Далее необходимо сопоставить извлеченные имена из разных статей друг с другом чтобы определять, что разные статьи относятся к одному и тому же человеку. Для этого необходимо привести имя человека к нормальной форме. Для этой цели используется библиотека pymorphy2. Затем полученные слова сортируются в алфавитном порядке. Разделение на имена и фамилии не используется, так как задача определения фамилий работает недостаточно хорошо, особенно для иностранных фамилий. К тому же, благодаря деловому стилю написания новстных статей, при первом упоминании человека как правило используется его полные имя и фамилия. Более того, упоминание в новостной статье фамилии без имени обычно используется в устойчивых словосочетаниях, например "пакет Яровой". Таким образом появляется возможность в траекториях отделить упоминания самого человека от упоминания связанного с ним устойчивого словосочетания.

Для более точного работы будущей модели, учитываются имена, извлеченные из первых двух предложений и только при условии, что

в этом же предложении нет других упоминаний имен. Это практически гарантирует то, что главным фигурантом новостной статьи будет именно тот человек, имя которого извлечено.

В дальнейшем для работы модели используется текст нескольких первых предложений, если в них не упоминается какая-либо другая личность. Суммарная длина этих предложений не должна превышать 200 слов (в среднем это 10 предложений). Это обоснованно тем, что суть новости обычно укладывается в эти 200 слов. Последующий текст чаще всего является уточнением или справкой о каких-то событиях или организациях. Также к этим предложениям добавляются те, в которых упоминается извлеченная в начале личность, если такие предложения есть в дальнейшем тексте.

1.3. Обзор данных

Единый формат, написать даты, дать оценку

Итоговый датасет представляет собой набор новостных статей сгруппированных по траекториям. Представленные в датасете СМИ: Лента, Дождь, Медуза, РИА Новости. СМИ Лента соответствует 450 траекторий состоящих из 31645 статей в сумме, средняя длина траектории 69.4. Данные СМИ Дождь это 405 траектория из 14136 статей средней длины 34.9. СМИ Медуза соответствует 63 траектории из 1618 статей, средняя длина 25.7. РИА Новости - 189 траекторий, 37058 статей, средняя длина траектории 196.1 Всего 1113 траекторий, 84457 статей, средняя длина траектории 75.9. Максимальная длина траектории - 500 статей.

2. Базовая модель

Базовая модель создана на основе модели **RMN**, представленной в статье Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships[1]

RMN решает следующие формализованные задачи. Первая из них — построить матрицу R размерности $K \times dim$, где K – задаваемое количество дескрипторов, dim – размерность векторного представления слов, состоящую из строк векторных представлений слов, которые и будут дескрипторами модели. Вторая — сопоставить каждой статье с номером t из траектории веса дескрипторов (распределение на дескрипторах), представленные вектором d_t размерности K.

2.1. Описание модели

Модель получает на вход векторное представление $\hat{u}_t$ новостной статьи

$$\hat{u}_t = \frac{1}{|\hat{S}_t|} \sum_{w \in \hat{S}_t} u_w \tag{1}$$

где t — номер текущей статьи в траектории, S_t — множество всех слов траектории, $\hat{S}_t$ — множество слов w текущей статьи, в которое каждое слово попадает с вероятностью p — параметр word dropout[5], u_w — векторное представление слова.

Word dropout используется для того, чтобы сделать модель более устойчивой. Также при использовании word dropout модель лучше выделяет долгосрочные тренды при использовании рекуррентного слоя.

Далее модель вычисляет вектор скрытого состояния

$$h_t = \mathrm{ReLU}(W_h \cdot u_t) \tag{2}$$

где $\mathrm{ReLU}(x) = \max(0, x)$ — функция активации, W_h — матрица весов

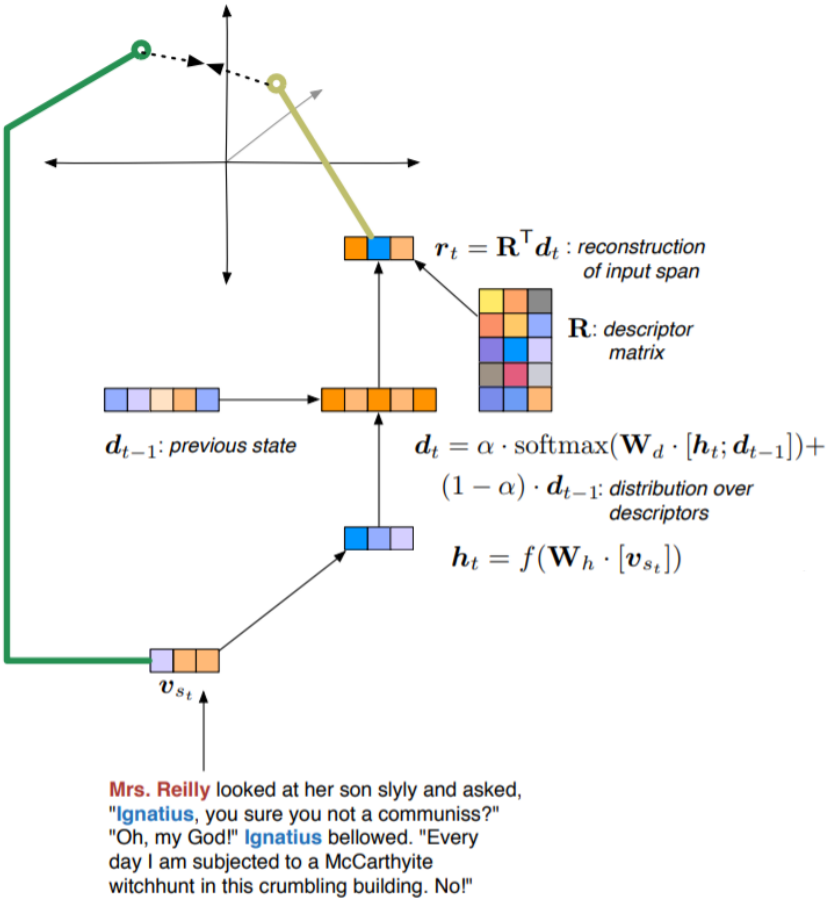

Рис. 1. Схематичное изображение базовой модели

слоя.

Затем с помощью рекуррентного слоя RNN с использованием результата предыдущего шага вычисляется текущее распределение на дескрипторах d_t

$$\begin{aligned} d_t &= \alpha \cdot \mathrm{softmax}(W_d \cdot [h_t; d_{t-1}]) + (1-\alpha) \cdot d_{t-1} \quad t > 1 \\ d_1 &= \mathrm{softmax}(W_d \cdot [h_1; \vec{0}]) \end{aligned} \tag{3}$$

где $\alpha \in (0, 1]$ — параметр сглаживания, $\mathrm{softmax}(X)_i = \dfrac{\exp(x_i)}{\sum\limits_{x_j \in X} \exp(x_j)}$.

Таким образом сумма всех компонент дескриптора равна единице.

Далее модель вычисляет вектор-реконструктор

$$r_t = R^T d_t \tag{4}$$

Матрица R нормированная и обучаемая. Задача вектора-реконструктора — приближать начальный вектор $u_t = \frac{1}{|S_t|} \sum_{w \in S_t} u_w$ новостной статьи, который был вычислен без использования word dropout.

Таким образом матрица R состоит из строк, которые можно интерпретировать как векторные представления слов. Эти слова можно выявить, найдя ближайшие к строкам матрицы векторные представления слов используя косинусное расстояние. Вектор d_t отражает на сколько каждое из этих слов описывает исходную новостную статью. За счет параметра α обеспечивается гладкость распределения на дескрипторах во времени

Функция ошибок

$$L(\Theta) = J(\Theta) + \lambda X(\Theta) \tag{5}$$

состоит из двух слагаемых.

Первое слагаемое

$$J(\Theta) = \sum_{n \in N} \max(0, 1 - r_t \cdot u_t + r_t \cdot u_n) \tag{6}$$

минимизирует косинусное расстояние между вектором новостной статьи и вектором-реконструктором. Также эта функция максимизирует это расстояние между вектором-реконструктором и случайно выбранными N векторами u_n (negative sampling[6]). Таким образом модель принуждается обучать более уникальные дескрипторы для каждой статьи.

Второе слагаемое

$$X(\Theta) = ||RR^T - I|| \tag{7}$$

отвечает за ортогональность матрицы дескрипторов R. Это означает, что вектора дескрипторов должны быть как можно более отдаленными друг от друга в смысле косинусного расстояния. λ – параметр ортогональности.

В базовой конфигурации используется значение word dropout p = 0.75, $\alpha = 0.5$, N = 50, коэффициент скорости обучения lr = 10e-3, параметр ортогональности λ = 10e-4. Оптимизация модели осуществляется методом Adam[7].

2.2. Достоинства и недостатки

Достоинством модели является то, что она не только выделяет тематики траекторий во времени, но и находит дескрипторы, которые должным образом кластеризируют блоки траекторий. Более того, эта модель не нуждается в обучающем множестве.

Несмотря на свои достоинства, модель имеет недостатки, связанные с ее архитектурой.

Задача параметра α - сглаживать траекторию, то есть обеспечивать незначительное отклонение текущего распределения на дескрипторах от предыдущего. Это обеспечивает выделение в качестве дескрипторов более долгосрочных тематик. Данный способ кажется слишком "жестким". К тому же, есть все основания полагать, что функция ошибки будет меньше при α близком к 1, но в этом случае распределение на дескрипторах будет недостаточно гладким. Возникает желание сконструировать такую сеть, в которой уменьшение значения функции ошибок будет однозначно отражать то, что модель лучше справляется с поставленной задачей.

Рекуррентный слой RNN является устаревшим и имеет ряд недостатков(можно сослаться на работу про проблемы рнн). К настоящему времени существуют архитектуры, которые решают проблемы RNN и работают сравнительно лучше.

3. Модернизация базовой модели

В данной главе описываются изменения базовой модели

3.1. Изменение рекуррентного слоя

3.1.1. Идея

Рекуррентный слой RNN имеет ряд недостатков, к примеру затухающие и "взрывающиеся" градиенты, проблемы с обработкой долговременных зависимостей.

Внутри рекуррентных сетей GRU и LSTM в качестве функций активации используются сигмоида и гиперболический тангенс, область значений которых лежит в промежутке (0, 1) и (-1, 1) соответственно.

Более того, сеть LSTM имеет два внутренних состояния. Одно из них отвечает за краткосрочную память, другое — за долговременную. Благодаря этому LSTM эффективно решает задачи, в которых возникает необходимость обрабатывать долговременные зависимости.

3.1.2. Результаты

Для сравнения работы различных вариантов модели используются следующие конфигурации сети.

Векторное представление (embeddings) слов:

1. Word2vec обученный на датасете Russian National Corpus, размерность 300
2. FastText обученный на текстах Википедии и Lenta.ru, размерность 300
3. FastText обученный на текстах Twitter, размерность 100

Рекуррентный слой:

embedding	alpha	RNN		GRU		LSTM	
		loss	α	loss	α	loss	α
word2vec	0.5	0.7384		0.7396		0.7575	
	trained	**0.7269**	0.93	0.7287	0.93	0.7306	0.99
fastText wiki+lenta	0.5	0.7107		0.7133		0.7292	
	trained	**0.7002**	0.91	0.7013	0.91	0.7027	0.99
fastText twitter	0.5	0.7470		0.7476		0.7651	
	trained	**0.7345**	0.9	0.7352	0.9	0.7371	0.99

Таблица 1. Результаты работы модели

1. RNN
2. GRU
3. LSTM

Параметр сглаживания α:

1. 0.5
2. Обучаемый параметр (trained)

При анализе работы моделей с различными рекуррентными слоями стоит учитывать что более сложные рекуррентные слои требуют больше эпох для обучения, поэтому при фиксированном количестве эпох значение функции ошибки вероятно будет выше у сложного слоя, при условии что этот слой не улучшает работу модели.

Из полученных данных (Таблица 1) можно сделать следующие выводы:

1) Улучшенные рекуррентные слои не улучшают работу модели При обучении приближается к 1, то есть исходный вектор намного лучше приближается за счет текущего вектора даже несмотря на word dropout 0.75

2) Чем сложнее рекуррентный слой, тем больше α. Это можно объяснить то, что необходимая для работы модели информация лучше передается через внутреннее состояние сети чем через "жесткое" смешение распределений текущего и предыдущего дескриптора

3) Значение функции ошибки значительно зависит от способа векторного представления слов

Главная задача параметра α — сглаживать траекторию, то есть обеспечивать незначительное отклонение текущего распределения на дескрипторах от предыдущего. Это обеспечивает выделение в качестве дескрипторов более долгосрочные тематики. Но исходя из полученных результатов, можно выделить проблему касающуюся того, что "поведение" этого параметра желательно как-то зафиксировать, но при этом иметь больше свободы при обучении сети. Также возникает желание сконструировать такую сеть, в которой уменьшение значения функции ошибок будет однозначно отражать то, что модель лучше справляется с поставленной задачей.

3.2. Изменение функции ошибки

3.2.1. Идея

Логично переместить роль параметра α на функцию ошибок. Для этого изменим в исходной функции ошибок (5) первое слагаемое (6). Заменим вектор u_t, являющийся векторным представлением статьи на $\hat{u}_t$

$$\begin{aligned} \hat{u}_t &= \beta \cdot u_t + (1-\beta) \cdot \hat{u}_{t-1} \quad t > 1 \\ \hat{u}_1 &= u_1 \end{aligned} \tag{8}$$

Таким образом за гладкость траекторий отвечает параметр β, поэтому теперь функция ошибок отражает то, какой результат от нее

embedding	alpha	RNN		GRU		LSTM	
		loss	α	loss	α	loss	α
word2vec	1	0.7803		0.7831		**0.7673**	
	trained	0.7712	0.61	0.7735	0.61	0.7727	0.98
fastText	1	0.7520		0.7543		**0.7439**	
wiki+lenta	trained	0.7439	0.45	0.7445	0.45	0.7444	0.97
fastText	1	0.789		0.79		**0.7784**	
twitter	trained	0.7793	0.44	0.7794	0.44	0.7785	0.98

Таблица 2. Результаты работы модели

требуется. Это означает, что мы можем обучать параметр не переживая, что реальный результат модели от этого ухудшится.

Зафиксируем $\beta = 0.5$

3.2.2. Результаты

Используем ту же сеть конфигураций что и в предыдущем разделе, в котором анализировалась работу предшествующей модели (Таблица 1). Только теперь зафиксируем $\alpha = 1$, при этом значении выходной вектор рекуррентного слоя не смешивается с вектором предыдущего распределения на дескрипторах. Это необходимо для того, чтобы оценить, осталась ли необходимость использовать жесткое смешение весов дескрипторов для достижения гладкости.

Из полученных результатов (Таблица 2) можно сделать следующий важный вывод. При использовании рекуррентного слоя LSTM получены достаточно хорошие результаты, особенно учитывая то, что это самая долгообучаемая сеть из представленных. Более того, эти результаты были лучше тогда, когда параметр α был зафиксирован и равнялся единице, то есть распределение предыдущего дескриптора не смешивалось с текущим. Это означает, что мы можем отказаться от этого компонента, который ранее отвечал за гладкость траектории. Внутренних состояний слоя LSTM оказалось достаточно, чтобы

должным образом приблизить вектор $\hat{u}_t$.

Таким образом оптимальная конфигурация модели — LSTM с фиксированным $\alpha = 1$. В дальнейшем используется векторное представление слов fastText wiki+lenta.

3.3. Дополнительные изменения

4. Сравнение моделей

Оценивание результатов работы моделей — отдельная сложная задача. Использовать какие-либо формальные функционалы качества не представляется возможным в связи с тем трудно определить формальные требования к результатам работы модели.

Для сравнения результатов базовой модели и модели, полученной в данной работы был создан telegram-бот. Пользователю для сравнения поступает изображение с графическими результатами работы базовой и итоговой модели для одной и той же траектории. На графике для каждой модели изображены блоки тем траектории, определяемые как несколько статей, идущих подряд во времени, у которых дескриптор с максимальным весом совпадает.

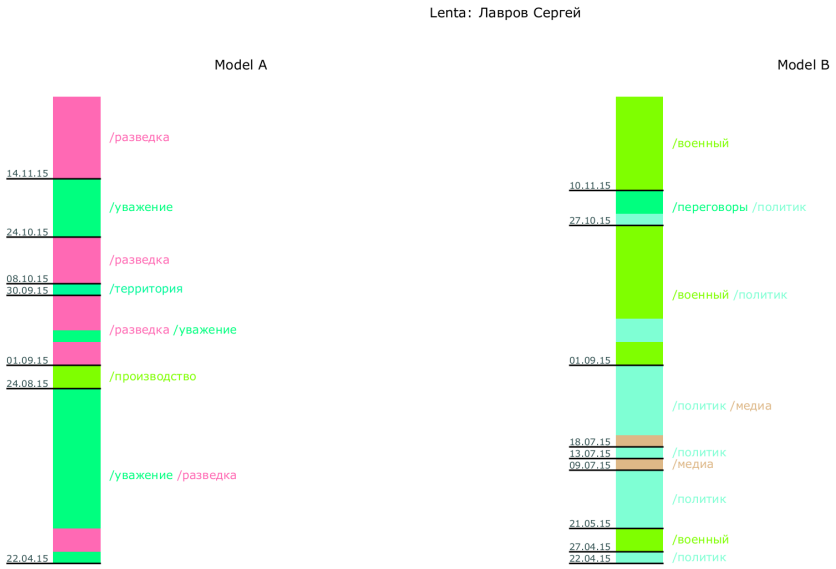

Рис. 2. Пример изображения

Выводы

Выводы

Заключение

Заключение

Список литературы

[1] Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships / Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi [и др.]

[2] Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. In Proceedings of Artificial Intelligence and Statistics

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. / Latent dirichlet allocation. Journal of Machine Learning Research, 3

[4] Jonathan Chang, Jordan Boyd-Graber, and David M Blei. 2009a. / Connections between the lines: augmenting social networks with text. In Knowledge Discovery and Data Mining

[5] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daume III. 2015. / Deep unordered composition rivals syntactic methods for text classification. In Proceedings of the Association for Computational Linguistics.

[6] Richard Socher, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. / Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics.

[7] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations.

[8] Lenta.ru [Электронный ресурс]: URL: `https://lenta.ru/` (дата обращения:).

[9] Tvrain [Электронный ресурс]: URL: https://tvrain.ru/ (дата обращения:).

[10] Meduza [Электронный ресурс]: URL: https://meduza.io/ (дата обращения:).

[11] РИА Новости [Электронный ресурс]: URL: https://ria.ru/ (дата обращения:).

[12] scrapy [Электронный ресурс]: URL: https://scrapy.org/ (дата обращения:).