Российский Университет Дружбы Народов

Факультет физико-математических и естественных наук

Кафедра информационных технологий

Курсовая работа на тему:

Глубокое обучение в задачах анализа текстов на естественном языке

Студент: Д. С. Ларионов
Научный руководитель: И. В. Смирнов

Москва, 2018

Содержание

1. Введение **2**

2. Обзор литературы **4**

3. Методология **6**

3.1. Эмбеддинги 6

3.2. Измерение качества предсказаний 6

4. Эксперименты **7**

4.1. Датасет 7

4.2. Препроцессинг 7

4.3. Классификаторы 7

4.3.1. Архитектуры Нейросетевых классификаторов 7

4.3.2. Гиперпараметры 8

4.4. Эмбеддинги 8

4.5. Метрики качества и оборудование 9

5. Результаты и обсуждение **9**

6. Заключение **9**

1. Введение

В течении последних 10-11 лет социальные сети прочно закрепились в нашей жизнию Люди привыкли обмениваться информацией, коммуницировать друг с другом практически круглосуточно. Исключением не являются и экстраординарные ситуации, происходящие в нашей жизни. Вполне естественно наблюдать всплески трафика социальных сетей во время событий, которые так или иначе затрагивают большое количество людей., например, во время всеобщих праздников, финалов спортивных мероприятий или черезвычайных ситуаций. И если первые два типа событий происходят ожидаемо, то черезвычайные ситуации, такие как теракты, землятресения или цунами, случаются внезапно.

Притом, практически всегда пользователи социальных сетей оказываются быстрее традиционных СМИ и представителей власти по части сообщения о факте существования той или иной ЧС. Разница по скорости реакции составляет от нескольких десятков минут до нескольких часов. Мы предполагаем, что анализ данных социальных сетей с помощью алгоритмов машинного обучения позволит фиксировать факт того, что то или иное ЧС произошло наиболее оперативно. В свою очередь, оперативное обнаружение таких ситуаций позволит оперативно принимать решения как органам местной/региональной власти, так ипростым людям. В частностиии, информация такого рода может быть полезна при планировании маршрута передвижений по затронутой территории.

Помимо масштабных ЧС, так или иначе затрагивающих город ил страну целиком, существует ещё один тип ЧС - локальные. Такие ситуации, как правило, затрагивают как отдельных людей, так и небольшие группы, но в целом они не влияютна остальных жителей региона. Локальные ЧС иногда вообще не освящаются ни в СМИ, ни в официальных пресс-релизах органов власти. Что вполне логично, нет смысла рассказывать о ситуациях которые не влияют на большое количество людей. Однако информация о ЧС такого масштаба может быть полезна, например, при расчете ценовых планок аренды ил продажи жилья в том или ином районе. В данном кейсе анализ данных социальных сетей является практически единственным способом получить информацию о факте проишествия.

Согласно исследованию [1], преобладающими типами контента, который пользователи публикуют в соцсетях, являются изображения и текст. Притом доля изображений в несколько раз больше доли текста. Однако в данной работе мы сфокусируемся на работе с текстовыми данными. Такой выбор сделан, в частности, из-за большого количества исследований, касающихся анализа текстовых данных социальных сетей. В работе будет представлен сравнительный анализ существующих алгоритмов машин-

ного обучения для задачи классификации текстов. Для тренировки будут использованы корпуса текстов из социальной сети Твиттер на английском языке. Анализ будет проведен по таким характеристикам как точность, вероятность ошибки.

2. Обзор литературы

В целом, сама идея того, что социальные медиа можно использовать как канал оперативной информации о происходящих черезвычайных ситуациях, уже давно интересует исследователей. Обьективная полезность такого канала была показана в [2]. В данной работе был проведен анализ нескольких крупных социальных медиа на предмет того, как полно данные описываются их пользователями и того, как оперативно эти описания появляются. В результате проведенных исследований было выяснено, что хоть и все анализируемые социальные медиа имели достаточно полное описание целевых событий, платформа Twitter охарактеризовала себя как наиболее оперативный источик информации. Поэтому, в нашей работе мы сфокусируемся на анализе Twitter-специфичных данных.

Хоть и задача анализа текста из твитов является, в общем, задачей анализа текста на естественном языке, мы должны учитывать "специфичность"того языка, на котором пользователи Twitter пишут сообщения(обилие сокращений, хэштегов, грамматических ошибок и т.д). Анализ способов применения различных алгоритмов машинного обучения для задачи классификации twitter-специфичного текста были представлены в работах [3, 4]. В этих исследованиях авторы анализирует различные техники, применяемые для задачи обнаружения событий. Список анализируемых техник включает в себя как и алгоритмы машинного обучения, так и статистический анализ. В первой работе авторы не проводят какие либо результаты экспериментов, но обозревают работы дргих исследователей по части используемых алгоритмов и наборов признаков. Во второй работе исследователи проводят эксперименты на пяти различных алгоритмах классификации и приводят метрики, полученные в результате. В заключении авторы называют два наиболее эффективных алгоритма: SVM и Random Forest. Однако те показатели, которые были получены на этих алгоритмах выглядят недостаточными для приемлемого функционирования анализа для наших задач. Мы намерены протестировать некоторые современные state-of-the-art модели на основен нейронных сетей(пример применения такой модели был показан в работе [5]) что бы получить исчерпывающий ответ на вопрос: какой алгоритм классификации текстов показывает лучшие результаты на twitter-специфичных данных.

Ещё одним интересным подходом к классификации текстов является pattern-based matching. Как показано в работе [9], алгоритм на основе сопоставления с образцовым набором фраз и слов может давать неплохие показатели точности и полноты, однако он не лишен и сивоих недостатков. В следствии того, что в нашей задачи нам гораздо важнее получить информацию из твитов, непосредственно связанных с происходящим событием, а не общим упоминанием подобного события, нам важно учитывать внутреннюю структуру текста. Pattern-based matching, по мнению исследователей

в этой работе, способен верно угадывать общий смысл исходя из того, какие слова используются в сообщении, однако он полностью игнорирует связь между словами. Авторы оригинальной статьи показывают, что алгоритм на основе машинного обучения выигрывают у него по показателю релевантности классифицируемых сообщения событию.

Рассмотрим существующие датасеты, подходящие для нашей задачи. Во-первых, нельзя не упомянуть семейство датасетов CrisisLexT6 [10]. На сегодня это самая объемная коллекция твитов, связанных с тем или иным черезвычайным проишествием. Для нашей задачи видится интересным датасет CrisisLexT6. Он содержит унесколько десятков тысяч твитов, которые в той или иной мере были связаны с громкими проишествиями в США и мире за период 2011-2013 годов. Каждый из твитов имеют метку о уровне связанности с основным проишествием. Однако, так как своей задачей мы ставим обнаружение не только громких и массовых черезвычайных ситуаций, но и меньшего размера, локальных, нам так же потребуется датасет для такого типа проишествий. Здесь наиболее интересным мы считаем датасет CloudFlower10k. Он содержит 10 тысяч твитов о ЧС различного масштаба: от лесных пожаров до небольших ДТП. Данные отобраны вручную и каждый твит идет вкупе с коэффициентом уверенности в том, что твит не является спамом и вообще содержит достоверную информацию о проишествии.

3. Методология

Пусть есть датасет из n документов D_i. Каждый документ представлен текстом твита и соответствующим классом. Классы в даной задаче - бинарные: документ может или относиться к ЧС или нет. Каждый из текстов представлен исходной строкой из twitter api, потому необходимо произвести препроцессинг для приведения текста к виду, пригодному для анализа. После, тексты должны быть преобразованы в вектора признаков $x_i \in \mathbb{R}^m$. Для этого мы используем такие техникик как эмбеддинги слов. Алгоритм эмбеддинга должен представить смысл каждого текста как элемент из векторного пространства. После извлечения признаков мы поделим датасет на две неравные части - для тренировки алгоритмов и для замера качества и проведем сравнительный анализ для некоторого набора алгоритмов классификации.

3.1. Эмбеддинги

Смысл алгоритма эмбеддинга - отразить смысл текста в виде элемента в векторном пространстве. Существуют два больших раздела, нак которые разделяются алгоритмы эмбеддингов: эмбеддинги слов и эмбеддинги предложений. В общем, идея у них одинаковая - натренировать нейронную сеть приближать разницу и схожесть слов и предложений.
В нашем случае, для того чтобы получить единый вектор смысла для отдельного документа мы будем брать средний от векторов каждого слова закодированных эмбеддингами слов. Для алгоритма эмбеддинга предложений мы будем подавать каждый текст как единое предложение, независисмо от того, есть внутри несколько предложений или нет.

3.2. Измерение качества предсказаний

Так как основной задачей в данной работе является выбор лучшей пары эмбеддинг-классификатор, мы должны уметь замерить качество работы для каждой пары. Пусть у нас будет разделение датасета на тренировочную часть X_{train}, Y_{train} и тестовую X_{test}, Y_{test}. После тренировки эмбеддинга и классификатора $f_i(x)$ на тренировочной части, мы используем модель для предсказания классов из тестовой выборки:

$$Y_{pred} = \{f(x_i) | \forall x_i \in X_{test}\}$$

После, мы подсчитаем значений функции качества $Q(Y_{pred}, Y_{test})$. Чтобы избежать получения оптимистически завышенных результатов, обусловленных особенностями распределения твитов в выборках, все замеры качества будут проведены на K-fold Кросс валидации. В финале, мы представим среднее и стандартное отклонение метрики качества для каждой пары.

4. Эксперименты

4.1. Датасет

Для наших замеров мы выбрали датасет CrisisLexT6. Он содержит 60000 твитов, так или иначе связанных с 6 различными крупными ЧС. Каждый твите промаркирован по тому, относится он к ЧС или нет. Кроме того, несколько твитов присутствуют без класса. Мы удалили их из датасета. Набор данных хорошо сбалансирован, потому дополнительных мер для достижения баланса классов не требуется. Для K-fold кросс валидации мы используем 5 как значение праметра K, следовательно, каждый из блоков будет содержать 12000 объектов.

4.2. Препроцессинг

Мы используем регулярные выражения для того чтобы очистить текст и объединить разрозненные хэштеги, ссылки и эмоджи. Для начала, текст каждого твита разбивается на токены с помощью NLTK TweetTokenizer. После мы используем пакет "re"из стандартной библиотеки Python для применения регулярных выражений.

4.3. Классификаторы

В наборе тестируемых алгоритмов классификации представлены как классические модели, так и нейросетевые классификаторы:

- Logistic Regression, предложенная разработчитками Liblinear [18] с интерфейсом в Scikit-learn.
- Random Forest, описанный в [17], реализованный в Scikit-learn.
- Gradient Boosted Decision Trees, реализованные в LightGBM [19].
- Fully-connected neural nework
- CNN for text classification, описанные в [21].
- C-LSTM, предложенные в [28].

4.3.1. Архитектуры Нейросетевых классификаторов

Полносвязанная нейронная сеть представляет собой простой двуслойный перцептрон с дропаут слоем посередине. Функции активации для первого и второго слоя - ReLU и Softmax соответственно. Сеть реализованна на фреймворке PyTorch

0.4.
Сверточная нейронная сеть для классификации текстов, предложенная Юном Кимом используется для классификации документов, где каждое словов представлено соотв. вектором. Первым слоем идут несколько несвязанных сверток с различными размерами фильтров. Полученные feature maps проходят через ReLU активация и склеиваются в один вектор. После, вектор проходит через полносвязный слой с Softmax активацией и на выходе мы получаем вероятности принадлежности к каждому из двух классов.
C-LSTM построена таким образом, чтобы извлечь последовательные контекстные признаке с помощью сверточного слоя и после найти долгосрочные зависимости с помощью LSTM слоя. После, ветора признаков проходят через два полносвязаных слоя с Tahn и Softmax активацией.

4.3.2. Гиперпараметры

Там, где не указано, мы использовали гиперпараметры, идущие по умолчанию для данного алгоритма.

- **Random Forest** Количество деревьев в лесу - 1000.
- **Gradient Boosted Decision Trees** Максимальная глубина дерева - 20, число ветвей - 11, learning rate - 0.05. Количество деревьев - 4000 с ранней остановкой через 200 итераций.
- **Fully-Connected Network** Размер скрытого слоя - 256, вероятность дропаута - 0.5. Тренировка проводилась в течении 10 эпох с ранней остановкой через три эпохи. Оптимизатор - Adam, функция ошибки - BinaryCrossEntropy.
- For **CNN for sentence classification** Размеры фильтров [3, 4, 5] с 512 выходными каналами на каждый. Дропаут 0.5. Настройки тренировки идентичны полносвязной сети.
- **C-LSTM** Размер свертки - 3 с 128 каналами, max pooling с ядром размера 2, LSTM hidden size - 80 и дропаут 0.1. Размер первого полносвязного слоя - 60. Настройки тренировки идентичны полносвязной сети.

4.4. Эмбеддинги

В тестируемом наборе эмбеддингов пристуствуют FastText [23], в варианте, натренированном на нашем датасете и на корпусе английской википедии; GloVe[24], натренированный на корпусе CommonCrawl и на корпусе твитов; Word2Vec[15], натренированный на нашем датасете. В качестве эмбеддингов предложений используется InferSent[14], натренированный на корпусе AllNLI.

4.5. Метрики качества и оборудование

Так как используемый датасет хорошо сбалансирован по классам, мы решили использовать Accuracy и F1-меру как метрики качества. Так же, нам кажется необходимым привести замеры для baseline, в которм всем элементам тестовой выборки проставлен класс 1, для сравнения с результатами.
Все эксперименты проведены на машине с GPU Nvidia Tesla K20, 64Gb оперативной памяти и 32 ядрами CPU.

5. Результаты и обсуждение

Результаты измерений представлены в таблицах 1-2. Ниже мы обсудим некоторые идеи, на которые мы натолкнулись в процессе экспериментов.

- В большинстве случаев, эмбеддинги натренированные на корпусе общей тематики показали себя лучше, чем натренированные на корпусе твитов. Нам кажется, это связано с тем, что лексикон, используемы при описании кризисных ситуаций отличается от твиттер-специфичного и более близок к общему. Также на результаты могло повлиять различие в размерах корпусов, на которых тренировались эмбеддинги.
- Эмбеддинги предложений лучше предсказывают смысл текста чем средний вектор эмбеддингов слов. Мы считаем это обоснованным, т.к взятие среднего сильно замазывает истинную картину смысла предложения. Более того, среднее не учитывает важность одних слов в одном части текста и неважность других. Однако высокая размерность выходного вектора эмбеддингов предложения делает невозмодным использование их в паре с некоторыми алгоритмами классификации.
- Нейросетевые классификаторы почти всегда обеспечивают более стабильное качество предсказаний в отличии от классических моделей(кроме lightgbm). Следовательно, мы будем получать стабильно хорошие результаты независимо от данных для тренировки и для теста.

6. Заключение

В ходе данной работы мв провели качественное сравнение различных пар эмбеддинг-классификатор и по результатам измерений, лучшей парой является Fasttext, натренированный на CrisisLexT6 и CNN for text classification. Стоит так же заметить, что полученные результаты на 2% превосходят по Acccuracy результаты измерения в одной из недавних статей по данному датасету [27]

Таблица 1. Accuracy

	FstMean	FstStd	FsWikiMean	FsWikiStd	GlCCMean	GlCCStd	GlTwtMean	GlTwtStd	W2VMean	W2VStd	InfStMean	InfStStd	Baseline
LogReg	0.8772	0.0690	0.8270	0.0757	0.8846	0.0476	0.8486	0.0613	0.8897	0.0573	0.8930	0.0439	0.5434
Random Forest	0.8733	0.0747	0.8298	0.0853	0.8762	0.0627	0.8452	0.0825	0.8776	0.0720	0.8975	0.0425	0.5434
GBDT	0.9113	0.0009	0.8912	0.0020	0.9256	0.0018	0.8912	0.0020	0.9145	0.0027	N/A	N/A	0.5434
FullyConnected	0.9027	0.0036	0.8599	0.0041	0.9162	0.0039	0.8718	0.0029	0.9058	0.0020	0.9092	0.0026	0.5434
CNN	**0.9392**	0.0033	0.9296	0.0034	0.9346	0.0027	0.9230	0.0014	0.9248	0.0019	N/A	N/A	0.5434
CLSTM	0.9153	0.0025	0.9159	0.0052	0.9170	0.0050	0.9088	0.0072	0.9191	0.0051	N/A	N/A	0.5434

Таблица 2. F1

	Fst Mean	Fst Std	FstWiki Mean	FstWiki Std	GlCC Mean	GlCC Std	GlTwt Mean	GlTwt Std	W2V Mean	W2V Std	InfSt Mean	InfSt Std	Baseline
LogReg	0.8743	0.0848	0.8256	0.0921	0.8861	0.0528	0.8514	0.0693	0.8899	0.0671	0.8945	0.0493	0.7041
Random Forest	0.8693	0.0952	0.8233	0.1119	0.8744	0.0741	0.8399	0.1052	0.8744	0.0887	0.8945	0.0493	0.7041
GBDT	0.9171	0.0009	0.8984	0.0014	0.9306	0.0013	0.8986	0.0023	0.9201	0.0026	N/A	N/A	0.7041
FullyConnected	0.9099	0.0037	0.8684	0.0039	0.9223	0.0037	0.8808	0.0028	0.9120	0.0021	0.9088	0.0021	0.7041
CNN	**0.9432**	0.0034	0.9343	0.0035	0.9389	0.0029	0.9272	0.0017	0.9296	0.0017	N/A	N/A	0.7041
CLSTM	0.9211	0.0022	0.9222	0.0036	0.9226	0.0061	0.9153	0.0049	0.9230	0.0055	N/A	N/A	0.7041

Список литературы

[1] Olga V. Muronets, Content of Social Networks: Trends and Patterns.

[2] Miles Osborne and Mark Dredze. 2014. Facebook, Twitter and Google Plus for Breaking News: Is There a Winner?. In ICWSM

[3] Atefeh F., and Khreich W. (2015), A Survey of Techniques for Event Detection in Twitter, Computational Intelligence, 31, 132–164, doi: 10.1111/coin.12017

[4] Alfredo Cobo, Denis Parra, Jaime Navón: Identifying Relevant Messages in a Twitter-based Citizen Channel for Natural Disaster Situations. CoRR abs/1503.05784 (2015)

[5] Tien Nguyen, Dat & Ali Al Mannai, Kamela & Joty, Shafiq & Sajjad, Hassan & Imran, Muhammad & Mitra, Prasenjit. (2016). Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks.

[6] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370. http://nlp.stanford.edu/ manning/papers/gibbscrf3.pdf

[7] Emadi, Noora Al et al. “QT2S: A System for Monitoring Road Traffic Via Fine Grounding of Tweets.” ICWSM (2017).

[8] Devyatkin D., Shelmanov A. (2017) Text Processing Framework for Emergency Event Detection in the Arctic Zone. In: Kalinichenko L., Kuznetsov S., Manolopoulos Y.

(eds) Data Analytics and Management in Data Intensive Domains. DAMDID/RCDL 2016. Communications in Computer and Information Science, vol 706. Springer, Cham

[9] To, Hien & Agrawal, Sumeet & Ho Kim, Seon & Shahabi, Cyrus. (2017). On Identifying Disaster-Related Tweets: Matching-Based or Learning-Based?. 10.1109/BigMM.2017.82.

[10] Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014). . In International AAAI Conference on Web and Social Media. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8091

[11] spaCy - Industrial-Strength Natural Language Processing; https://spacy.io/

[12] Software Framework for Topic Modelling with Large Corpora. Radim Řehůřek and Petr Sojka. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

[13] Distributed representations of words and phrases and their compositionality. T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean. Advances in neural information processing systems, 3111-3119

[14] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. arXiv preprint arXiv:1705.02364.

[15] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space

[16] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

[17] L. Breiman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9(2008), 1871-1874.

[19] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In Advances in Neural Information Processing Systems (NIPS), pp. 3149-3157. 2017.

[20] Vinod Nair and Geoffrey Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. ICML. 2010

[21] Yoon Kim. Convolutional Neural Networks for Sentence Classification. http://arxiv.org/abs/1408.5882

[22] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam. Automatic differentiation in PyTorch. NIPS-W. 2017

[23] P. Bojanowski*, E. Grave*, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 2017. Vol. 5. pp 135-146.

[24] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. 2015.

[25] Roy Chowdhury S, Purohit H, Imran M. D-sieve: a novel data processing engine for efficient handling of crises-related social messages. InProceedings of the 24th International Conference on World Wide Web 2015 May 18 (pp. 1227-1232). ACM.

[26] Zhang S, Vucetic S. Semi-supervised discovery of informative tweets during the emerging disasters. arXiv preprint arXiv:1610.03750. 2016 Oct 12.

[27] Li H, Caragea D, Caragea C, Herndon N. Disaster response aided by tweet classification with a domain adaptation approach. Journal of Contingencies and Crisis Management. 2018 Mar;26(1):16-27.

[28] Zhou C, Sun C, Liu Z, Lau F. A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630. 2015 Nov 27.