СРАВНЕНИЕ МЕТОДОВ СЮЖЕТНОЙ КЛАСТЕРИЗАЦИИ НОВОСТЕЙ

Воропаев П.М. (voropaev@phystech.edu)
Сопильняк О.А. (olga.sopilnyak@phystech.edu)

МФТИ, Москва, Россия

Abstract
In this paper, we analyze several approaches to news clustering in Russian. We use text embeddings based on tf-idf, publicly available pre-trained language model (BERT), and mutilingual embeddings pre-trained on parallel corpora (LASER). Agglomerative clustering and DBSCAN are used to get the final result. To evaluate the approaches, we use the winning solution of Telegram Data Clustering Contest and a small manually created corpus.
Keywords: clustering, tf-idf, laser, bert

Введение

Новостные ленты являются популярным способом поиска, чтения и анализа актуальных новостей. Для организации новостных лент применяется кластеризация статей по сюжетам, где в сюжеты объединяются похожие статьи, описывающие одно и то же событие и предоставляющие в целом совпадающую информацию. Размеченные по сюжетам тексты также можно использовать как корпуса поиска предложений-парафраз и заимствований, выделять по ним тренды, анализировать упоминания персон и организаций.

В последнее время активно развиваются языковые модели семейства BERT, и существуют исследования, посвященные их использованию для тематической кластеризации новостей [1], что не так актуально для прикладного применения, нежели сюжетная кластеризация. Чтобы ознакомиться с сюжетом, как правило, достаточно одной статьи из него. Чтобы ознакомиться с темой, необходимо изучить несколько сюжетов. Во многих работах, посвященных сюжетной кластеризации, основной упор делается на масштабируемости решения, но не на качестве, поэтому используются лишь классические алгоритмы. [2][3] К тому же, для русского языка не существует сравнений методов, включающих в себя модели, которые показывают state-of-the-art результаты на других NLP задачах. Также остается неисследованной возможность обучать вектора поверх предобученных языковых моделей, хотя для других распространенных NLP задач такой подход дает хорошие результаты.

В данной работе, проведенной как часть исследования предложений-парафраз и их поиска в похожих текстах, приводится сравнение нескольких методов решения задачи, основанных как на классических алгоритмах, так и на state-of-the-art языковых моделях. Мы исследуем возможность дообучения модели на синтетически сгенерированном корпусе и предлагаем успешный вариант композиции алгоритмов. Рассмотрены тексты на русском языке, однако те же методы можно применять для любых языков. Мы также публикуем собранный вручную корпус, пригодный для базового тестирования алгоритмов кластеризации[1].

1. Постановка задачи и корпус

На вход подается моноязычный набор текстов с метаданными: дата и время публикации, источник публикации, заголовок и, опционально, ключевые слова (теги), короткое описание, автор и др. Необходимо кластеризовать данный набор по сюжетам. Хотя кластеризация не подразумевает жестко заданного эталона, предполагается, что в одном кластере будут содержаться новости, описывающее одно событие. Характерной особенностью такой кластеризации является большой объем входных данных и большое количество кластеров, нередко состоящих из 1–2 документов.

В ноябре–декабре 2019 года Telegram проводил соревнование, включавшее эту задачу для русского и английского языков. Доступны большие неразмеченные корпуса текстов и результат работы решений участников на тестовой выборке, поэтому для сравнения мы используем решение победителя. При ручной оценке случайного

[1] Дополнительные материалы (код и данные) доступны в репозитории: https://github.com/voropz/news-clustering

подмножества кластеров, полученных победителем, значительных ошибок обнаружено не было, но в корпусе много шума: малозначимых статей, публицистики или перепечаток старых текстов, не формирующих сюжеты.

Для дополнительного тестирования был собран маленький ручной корпус, содержащий московские новости за один день. Предполагается, что тематическая близость текстов усложнит задачу. В корпусе 171 текст и 68 сюжетов.

2. Алгоритм

Все рассматриваемые алгоритмы будут действовать схожим образом: для каждого входного текста строится векторное представление, затем полученные векторы обрабатываются одним из стандартных методов кластеризации. В ходе обработки могут применяться дополнительные эвристики.

2.1. Текстовые признаки

В исходном корпусе тексты на разных языках смешаны, поэтому используем fasttext[4][5], чтобы определить язык и выбрать русский. Далее при необходимости сегментируем тексты на предложения с помощью rusenttokenize. Слишком короткие предложения и предложения, содержащие ссылки и стоп-паттерны типа «Источник:», «Фото:» отбрасываются.

2.1.1. TF-IDF

Классический способ векторизации текста. Текст новости вместе с заголовком лемматизируется с помощью pymorphy2[6] или deeppavlov Neural Morphological Tagging (выбирает лучший вариант pymorphy согласно предсказанному тегу, использовался в сравнении). Словарь строится с порогом отсечения редких слов 0.001, что дает размерность вектора 8000. Далее векторное пространство сжимается SVD до 300.

2.1.2. LASER

Language-Agnostic SEntence Representations (LASER)[7] — модель от Facebook, обученная на параллельных мультиязычных корпусах. Строит языконезависимый семантический эмбеддинг предложения, что может позволить объединять в сюжеты новости на разных языках. Поскольку модель обучена на предложениях, на вход подается не весь текст, а только заголовок вместе с 3–4 предложениями из начала статьи. Эксперименты с усреднением эмбеддингов предложений всего текста показали худший результат, а в начале статьи содержится достаточная информация. Модель используется без дообучения.

2.1.3. BERT

Представители семейства языковых моделей BERT являются state-of-the-art для многих задач NLP. Мы использовали предобученную модель для русского языка[8]. Семантическое различение предложений не было целевой задачей при обучении, и модель показала плохое качество при использовании напрямую. Для улуч-

шения качества мы исследовали способы дообучения модели на целевой задаче в условиях отсутствия обучающей разметки. Результат оказался отрицательным, поэтому опишем только один из похожих вариантов: выход BERT (batch size, seq.len., 768) подаётся двусторонней LSTM размерности 2x128, выход которой проектируется полносвязным слоем в размерность 64. Обучение происходит с помощью triplet loss классификатора[9], веса BERT заморожены. Обучающий корпус генерируется автоматически из общего набора текстов. Негативные примеры подбираются из новостей того же издания за предыдущие дни с использованием эвристики похожести по LASER. Положительный пример составляется из самой новости: для одной части берем первые несколько предложений, для другой – заголовок и следующие после первой части предложения. Части подбираются сопоставимой длины.

2.1.4. Усреднение моделей

При анализе ошибок лучших моделей – TF-IDF и LASER – было обнаружено, что их ошибки слабо коррелируют. LASER хорошо определяет близкие предложения, но из-за ограничения на входную длину плохо работает на длинных статьях. TF-IDF может ошибиться на сильно перефразированных статьях, зато обрабатывает сразу весь текст. Мы предлагаем простейшее ансамблирование: нормировку и усреднение евклидовых расстояний TF-IDF и косинусных LASER. Так было получено лучшее качество.

2.2. Ключевые слова

Ключевые слова или категории, полученные в источнике, нельзя использовать напрямую, т.к. разные издания имеют разную категоризацию, теги расставляются бессистемно, некоторые издания не проводят категоризацию вовсе.

Другой вариант — автоматическое получение ключевых слов из текста. Однако выбор ключевых слов является нетривиальной задачей. Так, именованные сущности не подходят для решения задачи. Был проведен эксперимент с NER (реализация deeppavlov BERT-based, F1 = 98.1% [8]) для бинарной классификации «лежат ли в одном сюжете» для всевозможных пар текстов двух изданий. Классификатор отмечал пару положительной, если пересечение извлеченных из их текстов именованных сущностей было достаточно велико. Перебором порогов получена F1-мера 0.32 с невозможностью установить порог для получения высокой точности при ненулевой полноте. Извлеченными сущностями чаще всего оказывались «Путин», «Собянин», «Медведев», «Москва» и т.п. Для качественной кластеризации нужно также извлекать неименованные сущности и факты.

2.3. Дата и время

В потоке новостей часто попадаются синтаксически сходные тексты, относящиеся к разным событиям. Типичными примерами являются прогнозы погоды, статистика заболеваемости коронавирусом и сводки ДТП, формируемые из заготовок подстановкой данных. Такие тексты разделяются временем публикации и, если доступно, регионом публикации.

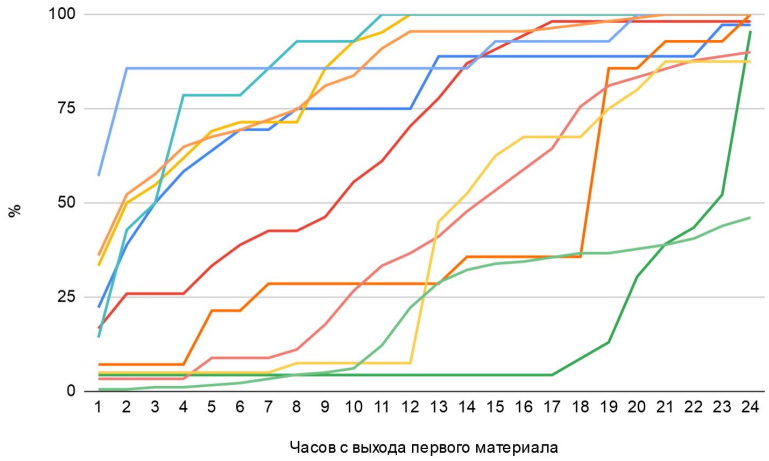

Рисунок 1 — Время публикации текстов 11 случайных сюжетов. Источник: Яндекс.Новости

Тексты одного сюжета обычно выходят в течение суток. На графике видны характерные скачки в момент публикации ведущими изданиями, происходящими в интервале от нескольких минут до 3 часов. Единственный многодневный сюжет, на наш взгляд, должен был быть разбит на несколько, т.к. содержит сообщение о самом событии, официальные комментарии и дальнейшее развитие. Иногда встречаются примеры поздних перепечаток (https://tass.ru/moskva/8383097 и https://guberniya.info/moscow/natalya-sergunina-rasskazala-kakie-m.html). Перепечатки осуществляются малоизвестными изданиями и, вероятно, носят рекламный характер.

Простейшим вариантом решения является разрезание входного набора новостей на отдельные дни, что дополнительно значительно снижает вычислительную сложность кластеризации. Правильнее использовать скользящее разрезание или добавлять удаленным по времени текстам штраф в расстояние. Порог удаленности и штраф являются гиперпараметрами алгоритма. Поскольку в корпусе все новости собраны за один день, эвристика не используется.

2.4. Источник

Маловероятно, что в одном кластере должны находиться новости из одного источника. Если это произошло в результате дублирования текста из-за технической ошибки или незначительного редактирования изданием, тексты будут иметь околонулевое расстояние в эмбеддингах по тексту и могут быть отфильтрованы.

Если в кластере находятся два различных текста одного издания, это может свидетельствовать об объединении нескольких сюжетов в один кластер. В этом случае предлагается продолжить кластеризацию, разбивая полученный кластер на более мелкие, пока в них не останется достаточно далеких (по эмбеддингам) текстов из одного источника. Такие тексты естественно выбрать новыми центрами подкластеров, либо сразу использовать иерархическую кластеризацию.

2.5. Кластеризация

Используется агломеративная кластеризация по евклидовому или косинусному расстоянию с настраиваемым порогом и различными методами вычисления межкластерных расстояний. На предварительных экспериментах DBSCAN показал в среднем худшие результаты.

3. Результаты

3.1. Корпус Telegram

Рассматривается корпус соревнования Telegram с новостями за 14 февраля (11445 текстов). Для оценки качества используем Homogeneity score, Completeness score и V-меру. В эталонном решении 6550 сюжетов и 1684 кластеров (сюжетов неединичного размера). Для испытываемых алгоритмов подбирались схожие показатели.

- TF-IDF без сжатия SVD, метод Уорда по евклидову расстоянию с порогом 1.
- TF-IDF с SVD, метод Уорда с порогом 0.4.
- LASER, метод средней связи по косинусному расстоянию с порогом 0.145.
- BERT, метод полной связи по косинусному расстоянию с порогом 0.03.
- Averaged TF-IDF + LASER, метод полной связи с порогом 0.5.

	H	C	V	кластеров
Winner vs TF-IDF svd	0.82	0.60	0.69	2804
Winner vs TF-IDF	0.67	0.77	0.72	1928
Winner vs LASER	0.71	0.74	0.72	1692
Winner vs BERT	0.61	0.45	0.52	1837
Winner vs Averaged	0.83	0.75	0.79	1787

Ручная проверка результатов TF-IDF и LASER показала в целом высокую похожесть текстов выделенных алгоритмами кластеров. Кластеры LASER казались полнее, хотя объективно оценить полноту вручную невозможно. На синтетических тестах с синтаксически схожими структурами (ДТП, погода) алгоритмы испытывали затруднения. LASER лучше разделял события и дальнейшие развития («данные утекли из банка», «банк опроверг утечку данных»).

3.2. Ручной корпус

171 текст, 47 кластеров и 68 сюжетов.

- TF-IDF, метод Уорда с порогом 0.44.
- LASER, метод полной связи по косинусному расстоянию с порогом 0.195.
- BERT, метод полной связи по косинусному расстоянию с порогом 0.13.
- Averaged, метод средней связи по косинусному расстоянию с порогом 0.48.

	H	C	V	кластеров
Reference vs TF-IDF	0.87	0.94	0.91	52
Reference vs LASER	0.90	0.98	0.94	45
Reference vs BERT	0.67	0.75	0.71	46
Reference vs Averaged	0.90	0.99	0.95	48

4. Выводы

Для поставленной задачи в условиях отсутствия обучающей выборки традиционный алгоритм TF-IDF успешно конкурирует с нейросетевыми подходами. SVD, особенно в малую (менее 300) размерность, скорее ухудшало качество, сливая в один кластер различные события на схожие темы, т.е. SVD больше подходит для тематической, а не сюжетной кластеризации.

LASER, несмотря на свою "архаичность"(LSTM encoder-decoder, не использует Transformer), подтвердил свою пригодность для обнаружения семантической близости. Модель можно использовать напрямую, дообучение необязательно.

BERT. Тяжелая модель демонстрировала нестабильность при дообучении и требует значительных ресурсов при применении. Обученная модель лишь незначительно превзошла необученную, хотя синтетическая задача во время обучения решалась уверенно. В полученных кластерах неизбежно оказывались совершенно неподходящие тексты. Мы пришли к выводу, что использовать глубокие языковые модели в условиях отсутствия большой обучающей выборки нецелесообразно.

Список литературы

1. Stankevičius L., Lukoševičius M. Testing pre-trained Transformer models for Lithuanian news clustering // Proceedings of the IVUS 2020. — Kaunas, Lithuania, 2020. — URL: https://arxiv.org/abs/2004.03461.
2. Multilingual Clustering of Streaming News / S. Miranda [и др.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — Brussels, Belgium : Association for Computational Linguistics, 2018. — с. 4535—4544. — DOI: 10.18653/v1/D18-1483. — URL: https://www.aclweb.org/anthology/D18-1483.
3. Linger M., Hajaiej M. Batch Clustering for Multilingual News Streaming // Text2Story@ECIR. — 2020.
4. FastText.zip: Compressing text classification models / A. Joulin [и др.] // arXiv preprint arXiv:1612.03651. — 2016.
5. Bag of Tricks for Efficient Text Classification / A. Joulin [и др.] // arXiv preprint arXiv:1607.01759. — 2016.

6. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts. т. 542 / под ред. M. Y. Khachay [и др.]. — Springer International Publishing, 2015. — с. 320—332. — (Communications in Computer and Information Science). — ISBN 978-3-319-26122-5. — DOI: 10.1007/978-3-319-26123-2_31. — URL: http://dx.doi.org/10.1007/978-3-319-26123-2_31.

7. Artetxe M., Schwenk H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond // CoRR. — 2018. — т. abs/1812.10464. — arXiv: 1812.10464. — URL: http://arxiv.org/abs/1812.10464.

8. Kuratov Y., Arkhipov M. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language // CoRR. — 2019. — т. abs/1905.07213. — arXiv: 1905.07213. — URL: http://arxiv.org/abs/1905.07213.

9. Weinberger K., Saul L. Distance Metric Learning for Large Margin Nearest Neighbor Classification // Journal of Machine Learning Research. — 2009. — февр. — т. 10. — с. 207—244.