НОРМАЛИЗАЦИЯ ИМЕН СОБСТВЕННЫХ БЕЗ СЕМАНТИЧЕСКОГО РАЗБОРА

Корзун В.А. (sdernal2@gmail.com)
ABBYY, Москва, Россия
МФТИ, Москва, Россия

PROPER NAMES NORMALIZATION WITHOUT SEMANTIC PARSING

Korzun V.A. (sdernal2@gmail.com)
ABBYY, Moscow, Russia
MIPT, Moscow, Russia

Abstract

This paper deals with proper names normalization using attention-based approach which has achieved good performance on machine translation and syntax parsing. Recent results based on a word-level features and tokens as input/output. Instead of that we have tried attention-based approach on character level, generating names by symbol. We also have tried some other modifications such as Conditional Random Fields and planned to use Reinforcement Learning or Variational Inference. However we have achieved appropriate result on corpus of news in Russian.

key words: seq2seq, attention, nlp

1 Введение

Seq2Seq подходы [1] позволяют достигать высоких результатов в ряде задач: машинный перевод [2, 3], синтаксический разбор [4]. Суть этого подхода следующая: по входной последовательности объектов получить скрытое представление совокупности, а затем использовать его при генерации выходной последовательности. Стандартным примером является машинный перевод: на вход подается последовательность слов в предложении на одном языке, из которого далее получается скрытое представление - "смысл" предложения, используя который генерируется последовательность слов на другом языке. Обычно этот подход реализуется с помощью двух рекуррентных нейронных сетей: энкодера, генерирующего скрытое представление, и декодера, которому на вход это скрытое представление подается и тот генерирует выходную последовательность.

Данный подход в чистом виде имеет существенный недостаток, встречающийся на длинных последовательностях: чем длиннее последовательность, тем сильнее затухает градиент к началу последовательности и тем больше информации нужно передать в скрытом представлении.

Эту проблему призван решить механизм внимания [5]. Если сделать предположение, что перевод слова зависит лишь от небольшого контекста, а не от всего предложения, то для генерации очередного слова требуется информация только об этом контексте. Механизм внимания как раз использует эту логику. Суть его заключается в том, чтобы сохранить все выходы рекуррентной нейронной сети энкодера (а не только последний) и при генерации очередного объекта декодером брать взвешенную сумму этих выходов, тем самым используя только необходимую информацию.

В данной работе рассматривается решение задачи нормализации имен собственных, используя приведенный выше подход. Текущие известные парсеры имеют проблемы с согласованием частей сущностей, например имени и фамилии, а также согласованием сущности и контекста. В качестве примера для сравнения рассматривался морфологический анализатор MyStem [12].

2 Постановка задачи

Предположим, что имеются предложения с указанными в них именами собственными в некоторой форме. Требуется сгенерировать нормальную форму этих сущностей.

Для оценки качества как основные метрики будем использовать долю полностью совпавших сущностей с эталоном без учета служебных символов, а также среднее расстояние Левенштейна.

В качестве входных данных будем использовать символы окна контекста некоторой сущности, т.е. на вход сети будут подаваться символы самой сущности, а также некоторое число символов справа и слева от неё. Данный подход используется, чтобы уменьшить входную последовательность

для уменьшения требуемых вычислительных ресурсов, при этом использовав контекст данной сущности. Примером полезного контекста является прилагательное, стоящее обычно слева от сущности, которое может указать на её форму: род, число, падеж. Еще одним примером является глагол, который также может указать на падеж и помочь в разрешении неоднозначности, например в предложениях

Александра Иванова видели сегодня утром.
Александра Иванова видела его.

именованные сущности имеют различную нормальную форму: *Александр Иванов* и *Александра Иванова* соответственно. Отличить одну сущность от другой позволяет глагол *видеть*, стоящий в соответствующей форме.

В качестве признаков последовательностей использовались только эмбеддинги символов, обучаемые в ходе основного процесса обучения, т.е. не предобученные заранее. Далее рассмотрим используемые модели.

3 Модели сетей

Эксперименты проводились над тремя вариациями seq2seq:

- Без механизма внимания.
- С механизмов внимания.
- С дополнительным CRF на выходе.

3.1 Простой Seq2Seq

Для начала опишем самую простую модель без механизма внимания. Данная модель состоит из двух основных частей: энкодера и декодера. Энкодер состоит из слоя эмбеддингов, который принимая на вход очередной символ (а точнее его номер в словаре), возвращает его векторное представление в пространстве вещественных чисел некоторой размерности. Второй составляющей энкодера является рекуррентный слой, принимающий на каждом шаге эмбеддинг символа из входной последовательности и возвращающий некоторое состояние. В данной модели нас интересует только последний выход энкодера, теоретически содержащий информацию о всем контексте сущности. Далее последний выход энкодера передается в декодер, который в свою очередь является также рекуррентной сетью. На каждом шаге в декодер передается копия выхода энкодера, а на выходе имеем распределение вероятностей по символам.

Забегая вперед, отметим, что данный подход плохо применим к данной задаче, т.к. фактически модель запомнила лишь сущности, на которых обучалась. Любая другая сущность, не участвующая в обучении, превращалась в несвязный набор символов, хотя отдаленно напоминающий исходную сущность. Однако данный подход неплохо себя показал в нормализации дат.

3.2 Seq2seq with Attention

Логическим развитием обычного seq2seq является добавление механизма внимания, описанного во введении. Данная модель также состоит из энкодера и декодера, но имеет существенные отличия. В энкодере суть осталась та же, кроме того, что мы сохраняем выходы на каждом шаге рекуррентного слоя. Декодер же в свою очередь принимает матрицу выходов энкодера, а также предыдущий сгенерированный символ. На каждом шаге декодер получает эмбеддинг предыдущего символа и, используя свое скрытое состояние и эмбеддинг, получает веса для выходов энкодера, взвешенная сумма которых и дает вектор, содержащий информацию о контексте, необходимом для генерации следующего символа.

Как ни странно, такой подход уже позволяет получить приемлемое качество, но также существуют ошибки, которые теоретически должна исправить следующая модель.

3.3 CRF

В предыдущей модели распределения вероятностей каждого символа получались независимо(хотя зависимости неявно были в рекуррентном слое), выбирался символ, имеющий наибольшую вероятность. Хотелось добавить в генерацию очередного символа зависимости от предыдущих сгенерированных символов. Одним из способов этого достичь, является добавлением к рекуррентному слою декодера слой CRF (условные случайные поля) [6, 7, 8], который учитывает предыдущие сгенерированные символы при генерации следующего, т.к. считает условную вероятность от предыдущего ответа.

Как ни странно, но добавление CRF не улучшило результат. О экспериментах и результатах речь пойдет далее.

4 Эксперименты и Результаты

Опишем процедуру подготовки данных. Предварительно весь текст приводится к нижнему регистру для уменьшения словаря, в котором хранится сопоставление символа и некоторого числа. Далее из текста выделяются сущности: контекст сущности и нормализованное значение. После этого сущности разбиваются на две непересекающиеся по нормализованным значениям группы, чтобы не тестироваться на объектах, участвующих в обучении. В итоге в сеть на вход подаются соответствующие символам числа из контекста, на выходе - из нормализованного значения.

В качестве корпуса использовались объединенные корпуса новостей Person-1000 [9] и размеченный корпус Лентапедии. Также проводилась фильтрация корпусов по следующим принципам:

- Расстояние Левенштейна между сущностью и её нормализованным значением не должно быть больше половины длины сущности, чтобы

отсечь нормализацию местоимений именами собственными

- Количество слов в сущности и её нормализованном значении должно совпадать (некоторые случаи удалось обойти удалением лишних слов из нормализованного значения)

В итоге получаем корпус со следующими характеристиками:

Число примеров	38854
Число уникальных сущностей	14848

Эксперименты проводились с использованием двух оптимизаторов: SGD и Adam, на которых результаты существенно отличались. Результаты работы обычного Seq2Seq рассматриваться не будут, т.к. фактически этот подход не работал. Ниже приведены результаты экспериментов для Seq2Seq с механизмом внимания, модели с дополнительным CRF, а также морфологического анализатора MyStem:

Результаты экспериментов

Model	Accuracy	Mean edit distance
Seq2Seq + Attn (SGD)	0.812	0.32
Seq2Seq + Attn (Adam)	0.905	0.12
Seq2Seq + Attn + CRF (SGD)	0.712	0.57
Seq2Seq + Attn + CRF (Adam)	0.877	0.18
MyStem	0.765	0.35

5 Анализ ошибок

5.1 MyStem

Как упоминалось во введении у морфологического анализатора MyStem наблюдаются серьезные проблемы с согласованием: как внутри сущности, так и между сущностями и контекстом. Разберем пример из постановки задачи:

Александра{александра} Иванова{иванов} видели{видеть}
сегодня{сегодня} утром{утро}.
Александра{александра} Иванова{иванов} видела{видеть} его{он}.

Из этого примера видно, что MyStem не отличает данные примеры друг от друга, а также выдаёт несогласованные части сущности. Далее приведем несколько простых примеров, где наблюдается несогласованность между частями сущности, а также некоторые другие ошибки:

Результат	Эталон
егор борисова, мария чернов, муртаза рахимов, джейкоб лить, воробей, анатолий сердюк	егор борисов, мария чернова, муртаз рахимов, джейкоб лью, воробьев, анатолий сердюков

Далее рассмотрим ошибки моделей на основе Seq2Seq подхода. А к примеру сравнения результатов сети и MyStem вернемся позже при пояснении карты внимания.

5.2 Seq2Seq with attention

Одним из основных типов ошибок является потеря части сущности (например фамилии) или же наоборот попытка интерпретировать глагол как часть сущности. Этот тип ошибок теоретически должен был поправить механизм, использующий функцию потерь на всю последовательность, например CRF. Также встречались ошибки разметки, ошибка в роде (потеря или лишняя *-а*) и просто искажение сущности (которое теоретически тоже должен был поправить CRF).

Примеры ошибок

Контекст	Эталон	Результат
...-министру дмитрию медведеву.	дмитрий медведев	дмитрий
...телю кпрф валерию рашкину.	валерий рашкин	валерий
...ны церкви святой евгении в...	святая евгения	свята вевгеии
...шофера хамзата тубурова...	хамзат тубуров	хамзата убурова

5.3 CRF

CRF смог превзойти предыдущую модель только с измененным оптимизатором. Из-за чего стало интересно проверить обычный Seq2Seq с attention при использовании этого оптимизатора.

5.4 Adam

Замена оптимизатора значительно улучшила результат и превзошла модель с CRF и этим оптимизатором. Собственно на данном этапе эта модель считается наилучшей и имеет следующие типы ошибок (Примеры будут записываться как "*контекст*: результат (эталон)"):

- Потеря части сущности. Значительно меньше чем раньше, всего один пример. Возможно проблема в формировании сущностей: контекст обрезается по предложению и получается два типа контекста: где справа от сущности есть еще слова и где лишь только знак препинания. Возможно из-за наличия таких двух типов у сети возникает некоторый диссонанс. Пример: *...енный еще виктором ющенко.*: виктор (виктор ющенко)

- Ошибка в роде. Пример: *...американцу гате камскому...*: гате камской (гата камский)

- Несогласованность частей сущности: довольно частая ошибка, когда имя имеет правильную форму, а фамилия имеет форму как в исходном

предложении. Пример: *...резидента александра чалого и советни...*: александр чалого (александр чалый)

- Неправильная форма части или всей сущности. Отличается от предыдущего типа ошибок тем, что генерируется несуществующая форма, например, полностью отрезается окончание: *...и блогере алексее навальном и в целом...*: алексей навальн (алексей навальный),

- Ошибки разметки. Например, нормальная форма сущности имеет другой порядок слов от порядка в предложении: *...нты подал горлов александр, 1963 год...*: горлов александр (александр горлов)

5.5 Карта внимания

Также для анализа ошибок полезно построить карту внимания, в которой видны веса используемых декодером выходов энкодера при генерации очередного символа.

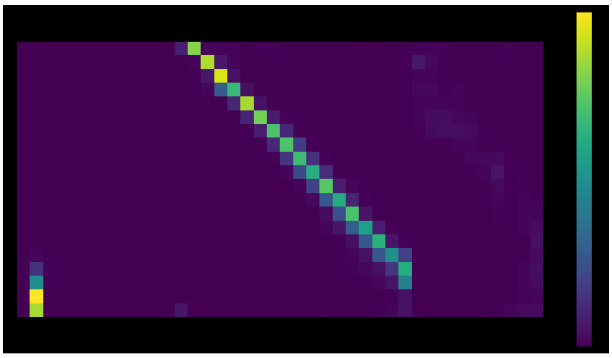

Рис. 1: Пример карты внимания

Из рисунка видно, что во внимание берется только узкий контекст, хотя в данном примере присутствует вначале глагол, окончание которого теоретически должно было помочь в определении формы. Скорее всего из-за того, что используется только узкая часть контекста, и присутствуют ошибки с определением рода. Для примера приведем вхождение одной сущности в разных контекстах:

Контекст	Эталон	Результат
...там была антонова...	антонова	антонов
...овый пост антоновой позволит...	антонова	антонова
...бщил, что антонова, которая...	антонова	антонова

Из этих примеров видно, что окончание *-ой*, характерное для женского рода правильно интерпретируется, а с окончанием на *-а* сложнее: наличие слева глагола женского рода не помогло в правильной генерации нормальной

формы, в отличие от прилагательного женского рода справа. Теперь рассмотрим результаты работы MyStem в режиме снятия омонимии на этих примерах:

там{там} была{быть} антонова{антонова}
новый{новый} пост{пост} антоновой{антонов} позволит{позволять}
сообщил{сообщать}, что{что} антонова{антонов}, которая{который}

Здесь наблюдается довольно странное поведение: в первом из трех примеров MyStem выдает правильный результат, в отличие от сети, но в остальных примерах ошибается, хотя второй пример кажется наиболее простым из-за окончанчания, соответствующего женскому роду.

6 Дальнейшее развитие

В качестве возможных модификаций в дальнейшем можно использовать механизмы из обучения с подкреплением (Reinforcement Learning), а также элементы нейробайеса - скрытые переменные.

6.1 Reinforcement Learning

Одной из целевых метрик является расстояние Левенштейна, поэтому кажется логичным оптимизировать эту метрику. Хотя использовать подход обучения с учителем для этой метрики не получится (функция дискретная и недифференцируема), можно использовать элементы обучения с подкреплением в качестве дополнения к функции потерь [10]. Суть этого подхода заключается в том, что помимо стандартной функции потерь, считающей логарифм вероятности правильной последовательности, можно использовать дополнительную, которая считается как

$$L_{rl} = (r(y') - r(y^s)) \sum_{t=\overline{1,n}} log(p(y_t^s | y_1^s, ..., y_{t-1}^s, x)),$$

где r - функция награды, y' - последовательность, полученная при жадной генерации (на каждом шаге выбираем наиболее вероятный символ), y^s - последовательность, полученная случайным образом по распределению вероятностей символов на каждом шаге. В качестве функции награды как раз можно использовать величину, обратную расстоянию Левенштейна до эталонной последовательности.

Данный подход теоретически позволит выбирать не только наиболее вероятные последовательности на каждом шаге, но также "отходить в сторону" , что позволит генерировать более правдоподобные последовательности, т.к. генерация наиболее вероятного символа на каждом шаге не гарантирует максимизацию вероятности генерации всей последовательности.

6.2 Latent variables

Еще одним из новых подходов в обучении нейронных сетей являются элементы нейробайеса, например, скрытые переменные. Суть подхода состоит в том, что объектная модель зависит от некоторых скрытых явлений и поэтому в модель нейронной сети добавляются дополнительные переменные, получающиеся из некоторого распределения. Пример использования скрытых переменных описан в статье [11], решающей задачу анализа тональности(Sentiment Analysis). Причем этот подход превосходит тот, что основан на механизме внимания.

В нашей же задаче можно попытаться применить скрытые переменные в двух местах: вместо механизма внимания брать переменные, например, из нормального распределения с центром в генерируемом символе и использовать их как веса для суммирования выходов энкодера, аналогично механизму внимания. Второй же возможностью является генерация случайной последовательности в ходе обучения с подкреплением: мы можем брать следующий символ не просто "подбрасывая монетку" , а выбирая из некоторого распределения.

Впрочем, приведенные выше идеи использования скрытых переменных не имеют логического подкрепления, почему они должны работать и имеет ли вообще смысл их использовать таким образом.

7 Заключение

Проведенные эксперименты показали, что Seq2Seq подход может показывать хорошие результаты не только на уровне слов, но также и на символьном уровне. Также удалось превзойти результаты морфологического анализатора MyStem при нормализации имён собственных, основными проблемами которого оказались несогласованность сущностей с их контекстом и между частями сущностей.

Также эксперименты показали, что более сложные модели могут работать хуже, чем простой Seq2Seq с механизмом внимания, хотя без самого механизма данная задача не решается. Причиной лучшей работы более простой модели может также являться относительно малый объем обучающих данных, а также некоторая их неконсистентность.

Итого, можно утверждать, что задача нормализации именованных сущностей достаточно успешно решается приведенным описываемым подходом.

Список литературы

[1] Ilya Sutskever, Oriol Vinyals, Quoc V. Le (2014), Sequence to Sequence Learning with Neural Networks, available at: https://arxiv.org/pdf/1409.3215.pdf

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio (2015), Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, avaliable at: https://arxiv.org/pdf/1406.1078.pdf

[3] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio (2015), Neural machine translation by jointly learning to align and translate, available at: https://arxiv.org/pdf/1409.0473.pdf

[4] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton (2015), Grammar as a Foreign Language, available at: https://arxiv.org/pdf/1412.7449.pdf

[5] Minh-Thang Luong Hieu Pham Christopher D. Manning (2015), Effective Approaches to Attention-based Neural Machine Translation, available at: https://arxiv.org/pdf/1508.04025.pdf

[6] Xuezhe Ma, Eduard Hovy (2016), End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, available at: https://arxiv.org/pdf/1603.01354.pdf

[7] Guillaume Lample, Miguel Ballesteros,Sandeep Subramanian, Kazuya Kawakami, Chris Dyer (2016) Neural Architectures for Named Entity Recognition, available at: https://arxiv.org/pdf/1603.01360.pdf

[8] Kai Hu, Zhijian Ou, Min Hu, Junlan Feng (2018) Neural CRF Transducers for sequence labeling, available at: https://arxiv.org/pdf/1811.01382.pdf

[9] Vlasova N.A., Suleymanova E.A , Trofimov I.V. (2014), Report on the Russian-language collection for the task of extracting personal names from texts [Soobshhenie o russkojazychnoj kollekcii dlja zadachi izvlechenija lichnyh imen iz tekstov], available at: http://ai-center.botik.ru/Airec/index.php/ru/collections/28-persons-1000

[10] Romain Paulus, Caiming Xiong, Richard Socher (2017), A deep reinforced model for abstractive summarization, available at: https://arxiv.org/pdf/1705.04304.pdf

[11] Tao Lei, Regina Barzilay and Tommi Jaakkola (2016), Rationalizing neural predictions, available at: https://aclweb.org/anthology/D16-1011

[12] Ilya Segalovich (2003), A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine, available at: http://cache-mskm909.cdn.yandex.net/download.yandex.ru/company/iseg-las-vegas.pdf