Разработка простого морфологического анализатора

Владислав Кораблинов

06.11.2019

1 Постановка задачи

Задачей данной практической работы является разработка простого морфологического анализатора для русского языка. Он представляет собой программу, которая принимает на вход текст и подает на выход для каждого слова из текста тройку *(слово из текста, лемма, частеречный тег)* в формате *слово{лемма=тег}*.

2 Решение с помощью словаря словоформ

2.1 Словарь словоформ

Простейший подход к решению данной задачи - использование словаря словоформ. Каждая запись в таком словаре представляет собой набор из слова в начальной форме, информации о его части речи и списка его известных форм. Таким образом, для ответа на запрос по слову нам достаточно найти его в одном из списков словоформ, после чего назначить ему в качестве леммы нормальную форму, которой соответствовал список, и ее частеречный тег.

Такой подход очень прост в реализации, а также легко расширяем - словарь можно легко дополнять информацией о новых словоформах. В то же время мы можем столкнуться с низкой полнотой такого словаря. И, конечно же, мы сталкиваемся с главной проблемой всей обработки языка - неоднозначностями. В данном случае мы столкнемся с двумя вида неоднозначностей: частеречной и леммы.

Мы можем попробовать бороться с неоднозначостью двумя способами:

- Брать первую найденную в словаре пару "лемма-тег". Такой способ зависит от словаря, поэтому его нельзя оценить.
- Найти все возможные пары "лемма-тег"и выбрать из них случайную. Для каждого слова мы получим вероятность угадать правильную пару, равную $\frac{1}{n}$, где n - количество найденных пар.

В качестве словаря воспользуемся словарем ODict. Он основан на словаре Зализняка и открыт для пополнения всеми желающими пользователями.

Приступим к тестированию наших методов. Здесь и далее будем приводить три числа - минимальное, среднее и максимальное значение полученной оценки на 5 датасетах.

Получаем следующие результаты:

	Мин.	**Среднее**	**Макс.**
Выбор первой пары "лемма-тег"	75.53	74.53	76.17
Выбор случайной пары	79.13	78.01	81.05

Видно, что выбор случайной пары действительно работает намного лучше, поэтому дальнейшие методы мы будем сравнивать с ним.

2.2 Статистическое снятие неоднозначности тега

Посмотрим на результат работы нашего текущего метода.

Потом{пот=S} пытаются{пытаться=V} петь{петя=S} все{все=S} вместе{вместе=ADV} начинают{начинать=V} песню{песня=S} вторую{второй=A} третью{третий=A} но{но=CONJ} ни{ни=ADV} одной{одной=S} закончить{закончить=V} не{не=ADV} могут{мочь=V}

Видим, что для некоторых слов, как, например, “потом” , в большинстве случаев следует выбирать другую пару "лемма-тег". Попробуем воспользоваться корпусной статистикой для разрешения неоднозначностей. Для этого нам потребуется корпус со снятой омонимией. После этого мы сможем посчитать для каждой словоформы частоту ее возможных разборов и выбрать наиболее частотный. Заметим, что такая жадная стратегия выбора является оптимальной при условии, что мы не пытаемся учитывать контекст.

Для подсчета статистики будем использовать корпус со снятой омонимией OpenCorpora и корпус Syntagrus с соревнования MorphoRuEval-2017. Объем первого корпуса составляет примерно треть от объема второго, поэтому мы сможем увидеть влияние объема текста на результаты.

Теперь, если для словоформы в словаре возникла частеречная неоднозначность, мы будем разрешать ее с помощью насчитанной статистики. Это дает следующие результаты:

	Мин.	**Среднее**	**Макс.**
OpenCorpora	87.98	87.06	88.61
OpenCorpora + Syntagrus	88.66	88.41	88.80

Статитистика действительно помогает снимать неоднозначности, при этом увеличение размера корпуса в 4 раза не дает большого прироста качества (скорее всего добавляется больше редких слов, но наиболее популярные хорошо описываются и меньшим корпусом). Далее будем везде пользоваться объединением корпусов.

2.3 Статистическое снятие неоднозначности тега

Снова посмотрим на результаты работы нашего алгоритма и увидим, что неоднозначность леммы все еще ее снимается плохо:

Кроме{кроме=PR} того{того=S} постоянная{постоянный=A} модификация{модификация=S} исходного{исходный=A} кода{кода=S} повышает{повышать=V} вероятность{вероятность=S} возникновения{возникновение=S} ошибок{ошибка=S} в{в=PR} программном{программный=A} коде{код=S} модели{модель=S}

Тогда попробуем поступать так. Если для словоформы встречаем неоднозначность в словаре, просто будем брать для нее наиболее популярную пару "лемма-тег"согласно статистике корпусов. В дополнение к этому методу будем искать в корпусах разбор для словоформ, отсутствующих в словаре. Получаем следующие результаты:

	Мин.	**Среднее**	**Макс.**
Снятие неоднозначностей	91.73	91.29	92.04
Снятие неоднозначностей + поиск для незнакомых словоформ	92.87	92.58	93.09

Результат существенно улучшился, причем заметный прирост дает поиск в корпусах незнакомых словоформ. Поэтому мы можем сделать смелое предположение, что мы можем решить задачу гораздо лучше с использованием в первую очередь корпусов. Действительно, при изучении файлов статистики видно, что для многих словоформ частоты достаточно велики, что позволяет надеяться, что такая статистика хорошо описывает реальное распределение использования словоформ в языке. Таким образом, мы подходим к другоме методу разметки.

3 Решение с помощью корпусной статистики

Теперь для каждой словоформы мы будем в первую очередь искать наиболее популярную соответствующую ей пару "лемма-тег"в корпусах, и только если такая словоформа там отсутствует, смотреть в словарь. Если словоформа отсутствует везде, то вернем в качестве леммы саму словоформу, а в качестве части речи - существительное, как наиболее частую. Такой подход дает следующие результаты:

	Мин.	**Среднее**	**Макс.**
Подход от корпусов	95.35	95.12	95,68.

Видим, что мы действительно снова значительно увеличили полученные оценки.

4 Предложение по обработке неизвестных слов

Наш метод все еще не умеет разбирать неизвестные словоформы:

– *«Варкалось. Хливкие шорьки пырялись по наве.»*
Л. Кэрролл (в пер. Д. Орловской)

– *«Варкалось{варкалось=S} Хливкие{хливкие=S} шорьки{шорьки=S} пырялись{пырять=V} по{по=PR} наве{наве=S}»*
Лемматизатор (в бессилии)

Для этого можно воспользоваться классическим способом, оценивающим статистику по суффиксам для известных словоформ. Для каждой словоформы из корпуса мы будем находить разность между ней и ее леммой. Более формально, пусть имеется словоформа S и ее лемма L. Найдем максимальный по длине префикс P словоформы S такой, что P является префиксом L. Тогда объектом нашей статистики будет пара $(S - P, L - P)$, где под вычитанием мы подразумеваем отбрасывание префикса. Таким образом, изначально для всевозможных пар значение статистики равно 0, а после каждого нахождения такой пары к значению ее статистики добавляется 1. Теперь, когда мы встречаем незнакомое слово, мы перебираем все его суффиксы (включая нулевой), и находим наиболее часто встречающуюся пару, в которой первый элемент равен этому суффиксу. Затем среди всех суффиксов мы выберем пару с максимальным значением.

К сожалению, из-за недостатка времени этот способ не был реализован.

5 Выводы

По итогам проделанной работы сделаем несколько выводов:

1. Неплохих результатов ($\approx 75\%$) можно достичь минимальными усилиями.
2. Открытых источников и простых алгоритмов достаточно, чтобы достичь качества выше 95%.
3. Корпуса со снятой омонимией содержат больше морфологической информации, чем словари словоформ.