Об одном статистическом методе пополнения морфологического словаря

Yet another statistical method for non-vocabulary word flexion prediction based on text corpora

Черненьков Д. М. (drcheren@gmail.com)

Московский институт электроники и математики, Москва, Россия

В статье предлагается алгоритм предсказания парадигм словоизменения несловарной лексики из текстовых корпусов. В основе алгоритма лежит ряд вероятностных моделей словоизменения, а также метод машиннообучаемого отбора и ранжирования объектов. В статье проведен анализ статистических свойств корпуса и результатов обучения модели.

Введение

Создание и пополнение морфологического словаря для флективных языков крайне трудоемко по причине сложности правил русской морфологии. Однако существует возможность полностью или частично автоматизировать этот процесс средствами анализа текстов, содержащих несловарную лексику.

Существует несколько методов создания автоматических морфологий. Требования к морфологическому анализатору зависят от задачи, для решения которой он применяется. Так, в [1] описаны методы морфологического анализа для поисковых систем. Основными требованиями в данном случае являются производительность и полнота анализа. Т. е. анализатор должен выдавать результат для максимального количества различных словоформ. При этом подробная информация, как правило, не требуется. Для основных задач информационного поиска достаточно получения нормальной формы из произвольной словоформы. В такого рода анализаторах словарь не является обязательным компонентом, и даже при его наличии производится анализ как словарных, так и несловарных словоформ. В [2,3] описаны варианты алгоритмов морфологического анализа, использующие индексы постфиксов, и набор правил словообразования.

Системы, в которых морфологический анализ является лишь промежуточной ступенью, за которой, как минимум, следует ступень синтаксического анализа [4,5], нуждаются в более подробной грамматической информации о каждой словоформе. В русском языке можно насчитать до 15 грамматических категорий, так или иначе выразимых морфологически, до 100 грамматических значений на часть речи (и соответственно, до 100 форм на слово) и свыше 2000 парадигм словоизменения [6]. В связи с этим, для полного, точного и подробного морфологического анализа русских словоформ необходим словарный алгоритм. Причем словарь должен содержать полную грамматическую информацию о каждом слове и каждой словоформе.

Большинство словарных морфологий для русского языка основано на грамматическом словаре А. А. Зализняка. Однако для применения анализаторов в практических целях необходимо пополнение словаря тем или иным методом. В [7,8] описаны методы автоматического и автоматизированного пополнения морфологических словарей с использованием корпусов текстов. Эти методы основаны на статистических свойствах языка. Основной проблемой, которая решаются в этих работах, является проблема неоднозначности разбора несловарных словоформ и необходимости выбора правильной гипотезы лематизации. В [7] предлагается группировать гипотезы различными методами и выбирать группы с наибольшей встречаемостью словоформ в корпусе. В [8] данный метод дополняется учетом статистических свойств парадигм, а также учет наиболее используемых словообразовательных префиксов.

Оба метода дают достаточно высокую точность. Однако в [7] выходным результатом алгоритма является набор гипотез, из которых по-прежнему необходимо выбирать верную, т. е. процесс пополне-

ния словаря автоматизирован лишь частично. В [8], с другой стороны, на выходе мы получаем лишь канонические формы для несловарных словоформ, в то время как для многих задач необходимо определить полную парадигму словоизменения.

Кроме того, при решении задачи снятия омонимии (при которой возникает схожая проблема неоднозначности морфологического анализа) используются методы анализа ближайшего контекста анализируемой словоформы [9]. В основе метода лежит предположение, что соседние слова с большой вероятностью связаны грамматическими отношениями, такими как согласование или управление. Данный метод можно применять и для выбора гипотез лемматизации несловарных словоформ.

1. Цели и задачи

Целью данной работы является объединение существующих методов анализа несловарной лексики и пополнения словарей для создания максимально точного алгоритма пополнения, требующего минимальной ручной обработки результатов. За основу берется математическая модель и алгоритм, описанные в [10]. Кратко этот алгоритм можно описать как последовательность шагов:

1. Выделить несловарные словоформы из корпуса
2. По каждой словоформе построить все возможные гипотезы лемматизации. Объединить построенные гипотезы в одно множество без дубликатов
3. Отфильтровать гипотезы по некоторому признаку
4. Кластеризовать гипотезы, выделив компоненты связности в биграфе гипотезы-словоформы
5. Из каждого класса по некоторому критерию выбрать одну или несколько наилучших гипотез

Для достижения поставленной цели необходимо решить ряд проблем:

1. Оценка качества и анализ результатов. Отсутствие достаточно крупного размеченного корпуса с несловарной лексикой. Кроме того, пока неясен критерий оценки точности анализа. В приведенных во введении работах используются различные критерии качества. Необходима выработка критерия качества в соответствии с основным назначением данной системы.
2. Большое число признаков, которые нужно учитывать при отборе гипотез. Затруднителен ручной подбор критериев отброса и ранжирования гипотез.
3. Большое количество парадигм в словаре порождает множество гипотез для каждой словоформы. Среди этих гипотез лишь небольшая доля верных (около одной тысячной). При попытке анализа признаков совокупность верных гипотез оказывается статистически незначимой.

Проблема отсутствия размеченного корпуса решается разделением множества лексем существующего словаря на генерирующее и валидационное подмножества (с образованием соответственно двух словарей, генерирующего и валидационного, с общим набором парадигм). При этом имитируется ситуация пополнения словаря: множество анализируемых словоформ включает в себя все словоформы, которые не входят в генерирующий словарь, но входят в валидационный. Далее, во всех местах алгоритма, где обычно используется полный словарь, теперь используется генерирующий. Назначение валидационного словаря — проверка сгенерированных гипотез (в силу описанного алгоритма выбора словоформ, правильный их разбор всегда можно определить из словаря).

2. Сокращение числа гипотез

В связи с большим числом парадигм в словаре (около 2700), на каждую несловарную словоформу генерируется несколько тысяч гипотез. Такое большое количество гипотез, во-первых, серьезно сказывается на производительности, а во-вторых, делает долю правильных гипотез очень низкой, что затрудняет процесс машинного обучения. Для снижения числа неверных гипотез предпринимаются следующие меры:

- Кластеризация парадигм. Если у парадигм A и B одинаковые части речи и наборы неизменяемых параметров, и при этом множество форм парадигмы A целиком входит во множество форм B, то эти две парадигмы объединяются, а при генерации гипотез используется только B (с большим числом форм). Этот метод позволяет сократить число парадигм, а значит и генерируемых гипотез, приблизительно вдвое. Поскольку гипотезы считаются эквивалентными, верные разборы не теряются.
- Фильтрация гипотез перед кластеризацией. Подробнее описана в [10].
- Отсечение гипотез в кластерах. Для каждого кластера составляется список всех словоформ, покрываемых его гипотезами. Из этих словоформ выбирается 3, наиболее часто входящие в корпус. Гипотезы, не покрывающие хотя бы одну из этих словоформ, удаляются из кластера. При этом отсеивается около 60 % неверных гипотез и около 5 % верных. Данная эвристика предложена в [13], где она используется для сокращения размера поискового индекса.

Даже с учетом этих мер число неверных гипотез в выборке превышает число верных более, чем в 100 раз. Построение качественного регрессора на такой выборке затруднительно.

3. Применение машинного обучения

Задачи фильтрации и выбора объектов некоторого типа по множеству признаков можно сформулировать как задачи машинного обучения при условии наличия подходящей обучающей выборки. Кроме того, необходимо количественное выражение всех признаков отбора. Если использовать метод деления словарей, описанный выше, можно построить достаточно большую выборку гипотез, помеченных как либо верные, либо неверные. При этом для фильтрации можно использовать алгоритмы классификации (разделение на 2 класса — верные и неверные гипотезы), а для выбора гипотез из кластеров — алгоритмы регрессии (генерация метки-действительного числа для каждой гипотезы и выбор из каждого класса гипотез с наибольшими значениями меток).

Однако при построении классификатора для фильтрации возникает следующая проблема: в обучающей выборке присутствует хотя бы одна верная гипотеза для каждой анализируемой словоформы, в то время как основное назначение фильтрации — отбросить опечатки и прочий «мусор», т. е. токены, в принципе не имеющие верного разбора. Поэтому на данный момент предлагается использовать для фильтрации старые критерии, полученные в [10].

Проблема выбора верной гипотезы из кластера не сводится непосредственно к задачам классификации, регрессии и кластеризации, которые решаются основными методами машинного обучения. Попытка классифицировать гипотезы на верные и неверные приводит к очень низкой точности, поскольку мощность класса неверных гипотез приблизительно на 2 порядка больше мощности класса верных гипотез. В результате точность обычных методов классификации в применении к этой задаче не превышает 3–5 %, что лишь немного выше случайных результатов.

Скорее, данная проблема выбора гипотез аналогична проблеме ранжирования, которая наиболее распространена в системах поиска информации. Отличие лишь в том, что задача ранжирования состоит в сортировке элементов в правильном порядке, а описываемая задача — в выборе наилучшего из них. В [14] описаны два подхода к ранжированию:

1. Парное ранжирование. Задача сравнения ранжируемых объектов сводится к задаче классификации. Элементами выборки для обучения классификатора являются пары ранжируемых объектов, пары разделяются на 2 класса: те пары, в которых первый объект лучше второго, и те, в которых второй объект лучше первого. Поиск наилучшего объекта тоже может легко осуществляться этим методом. Однако данный метод обладает рядом недостатков. Во-первых, далеко не все алгоритмы классификации гарантируют необходимые свойства функции сравнения (транзитивность и антисимметричность). Во-вторых, для получения удовлетворительной точности выбора необходима очень высокая точность классификатора. Так, для выбора верного элемента из 10 со средней точностью 50 % необходима функция сравнения, работающая с точностью не менее 97 %. С ростом числа ранжируемых элементов допустимый процент ошибки классификатора падает с экспоненциальной скоростью.
2. Списковое ранжирование. Выбирается оценочная функция, ставящая в соответствие каждому объекту действительное число — оценку. При ранжировании объекты сортируются в порядке убывания оценок. При выборе просто берется объект с наивысшей оценкой. Этот подход и использован в данной работе.

Предлагаемый метод выбора гипотез из кластера основан на алгоритме ранжирования, представленном в [15]. Выбор наилучшей гипотезы из списка производится функцией y = h(X),где X — конечное множество гипотез $X=\{x_1, x_2, \ldots, x_n\}$, y –гипотеза, $y \in X$.

Для оценки гипотез вводится скалярная функция f(x), где x — гипотеза. Конкретный вид функции в данном разделе не важен. Тогда функция h имеет вид:

$$h(\{x_1, x_2, \ldots x_n\}, f) = \underset{j}{\operatorname{argmax}} f(x_j), \quad j = \overline{1,n}$$

Задача состоит в нахождении функции f*, обеспечивающей максимальную точность выбора гипотез, т. е. если в произвольном кластере гипотез X верной является y, то

$$f^* = \underset{f}{\operatorname{argmax}} P(h(X, f) = y$$

Естественно, в распоряжении имеется лишь ограниченная случайная выборка пар $S=\{\{X_1,y_1\},\ldots \{X_m,y_m\}\}$, с помощью которой необходимо оценивать точность выбора и оптимизировать функцию f. Для этого вводится приблизительная оценка точности L(S, f). Тогда мы можем определить ожидаемую функцию $\hat{f} = \operatorname{argmax}_f L(S, f)$. По аналогии с предложенной в [15] оценкой логарифмического правдоподобия для ранжирования, предложим функцию для оценки точности выбора:

$$L(S,f) = \sum_{i=1}^{m} \log\left(\hat{P}(h(X_i) = y_i)\right),$$

где

$$\hat{P}(h(X_i) = y_i) = \frac{\exp\left(f(y_j)\right)}{\sum_{j=1}^{|X_i|} exp(f(-x_i^j))}.$$

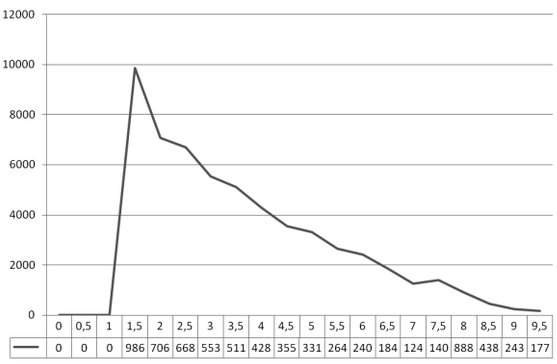
Эта оценка аналогична оценке правдоподобия, используемой в логистической регрессии. Согласно [16], такая оценка дает хорошее приближение, если распределение величины f относится к семейству экспоненциальных распределений. Наиболее распространенные в статистических задачах распределения, такие как Нормальное и биномиальное распределения, распределения Пуассона и Бернулли. Кроме того, функция L дифференцируема, что позволяет использовать градиентные методы для нахождения f.

4. Описание и анализ признаков

Для представления гипотез в функции оценки необходим набор количественных характеристик гипотезы. В [10] был предложен ряд признаков: частотность словоформ, покрываемых гипотезой, в корпусе, число различных покрываемых словоформ в корпусе, число лексем в словаре с тем же постфиксом псевдоосновы заданной длины. В данной работе предлагается модифицировать и расширить набор признаков. Ниже приводится полный список с описаниями:

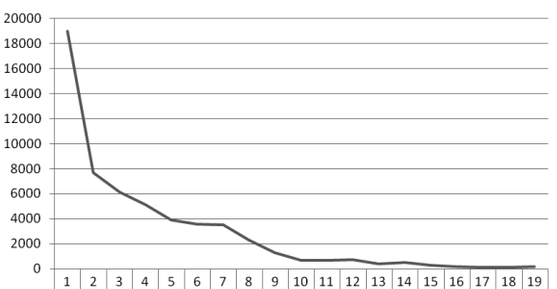
1. Число вхождений в корпус словоформ, покрываемых данной гипотезой. В качестве
2. Число различных словоформ входящих в корпус и покрываемых данной гипотезой
3. Число лексем в словаре с той же парадигмой, и имеющих тот же постфикс длины l, что и словоформы данной гипотезы, причем берется среднее арифметическое этих чисел для всех форм. Разным значениям l соответствуют разные признаки
4. Оценка вероятности вхождения грамматических форм в корпус. Пусть по гипотезе можно построить словоформы $w_1, w_2, \ldots w_n$. Им соответствуют грамматические значения $p_1, p_2, \ldots p_n$. (Грамматическое значение определяется частью речи и полным набором параметров лексемы и словоформы.) Значение данного признака считается по формуле $\sum_i F(w_i)\log\left(F(p_i)\right)$.
5. Оценка вероятности вхождения биграмм грамматических форм в корпус. Для каждого вхождения словоформ вычисляется частота вхождения в корпус биграмма грамматического значения этой словоформы и предшествующей ей в тексте словоформы. В качестве значения признака используется сумма логарифмов этих частот. Аналогичный признак вычисляется с учетом последующей словоформы, а не предыдущей.

Для исследования признаков был использован корпус новости с портала rbc.ru за период с января 2003 по декабрь 2008. Распределения значений признаков показаны на диаграммах приведенных ниже.



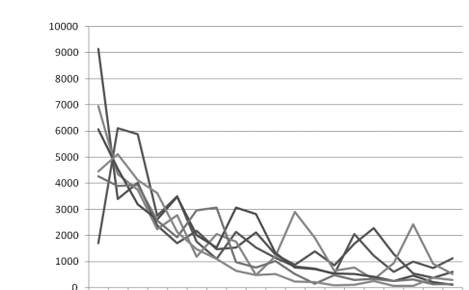
	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5
—	0	0	0	986	706	668	553	511	428	355	331	264	240	184	124	140	888	438	243	177

Рис. 1. Распределение частотности словоформ гипотезы



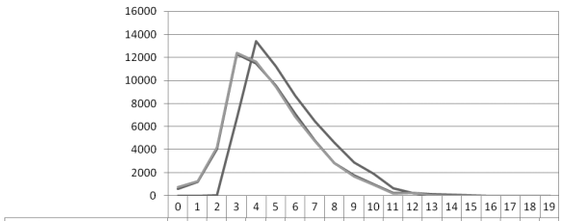
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	19	76	61	51	39	35	35	22	13	67	69	73	41	50	28	16	11	13	19

Рис. 2. Распределение числа встреченных словоформ гипотезы



	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
0 символов	17	61	58	27	35	17	11	21	15	11	82	72	52	20	12	60	10	74	11
1 символ	91	34	40	23	16	21	14	15	21	12	88	13	85	16	22	12	53	37	60
2 символа	69	43	37	22	27	11	20	17	49	12	29	19	65	77	32	94	24	91	52
3 символа	60	45	31	26	34	20	15	30	28	13	78	71	54	53	41	26	45	20	10
4 символа	42	38	39	25	19	29	30	96	77	10	52	15	48	29	32	24	31	12	12
5 символов	44	51	41	36	21	14	11	63	48	52	23	21	97	10	24	67	76	38	28

Рис. 3. Число лексем парадигмы с тем же постфиксом заданной длины



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Отдельные словоформы	0	0	68	67	13	11	86	64	46	29	19	65	25	14	86	37	1	0	0	0
Биграммы по префиксам	63	12	40	12	11	95	70	47	28	17	99	22	24	9	0	0	0	0	0	0
Биграммы по суффиксам	75	12	42	12	11	94	68	47	28	16	94	18	22	42	0	0	0	0	0	0

Рис. 4. Распределение оценок встречаемости отдельных словоформ и биграммов

5. Обучение и тестирование классификатора

В качестве функции f для ранжирования гипотез использовалась линейная комбинация признаков с коэффициентами θ. Для оптимизации значения коэффициентов был использован стохастический градиентный спуск [11]. Градиент рассчитывался как $\nabla_\theta L(S, f)$. Точность отбора в таком виде оказалось очень низким — 13 %.

Тогда был применен алгоритм пошагового отбора признаков. В результате качество повысилось до 37 %, а среди признаков осталось только три: число различных словоформ и вероятности биграмм (как с предшествующим, так и последующим словом).

Кроме того, была оценена вероятность нахождения правильных гипотез, если брать несколько гипотез:

Число отобранных гипотез	Вероятность найти верную гипотезу
1	37 %
2	49 %
3	55 %
4	59 %
5	60 %
6	60 %
7	61 %
8	62 %
9	62 %
10	62 %

Выводы

Разработан новый алгоритм предсказания модели словоизменения несловарной лексики из текстовых корпусов. Точность полностью автоматического отбора составляет 37 %.

Несмотря на то, что низкая точность не позволяет использовать данный алгоритм для полностью автоматического пополнения словаря, он позволяет существенно облегчить ручной метод заполнения. Аналогичных опубликованных результатов по заполнению словаря с точным соответствием полной парадигмы словоизменения найдено не было.

Отбор признаков показал, что наиболее важными критериями гипотез о словоизменении являются результаты анализа грамматического окружения вхождений несловарных словоформ. В дальнейших исследованиях необходимо провести анализ более дальнего окружения, чем соседние слова.

Литература

1. *Сегалович И. В., Маслов М. А.* Русский морфологический анализ и синтез с генерацией моделей словоизменения для не описанных в словаре слов // М.: Диалог, 1998.
2. *Гельбух А. Ф.* Эффективно реализуемая модель морфологии флективного естественного языка // М.: Всероссийский институт научной и технической информации, 1994.
3. *Ножом И. М.* Морфологическая и синтаксическая обработка текста (модели и программы), // М.: 2003,
4. *Крылов С. А., Старостин С. А.* Актуальные задачи морфологического анализа и синтеза в интегрированной информационной среде Starling // М.: Диалог, 2003.
5. *Мальковский М. Г., Старостин А. С.* Система морфосинтаксического анализа Treeton и мультиагентный синтаксический анализатор Treevial: принцип работы, система правил и штрафов // Екатеринбург: Изд-во Уральского университета, 2007. — С. 135–143.
6. *Зализняк А. А.* Грамматический словарь русского языка, 2-е изд. // «Русский язык», М.: 1980.
7. *Ляшевская О. Н., Сичинава Д. В., Кобрицов Б. П.* // Екатеринбург: Изд-во Уральского университета, 2007. — С. 118–125.
8. *Андреев А. В., Березкин Д. В., Симаков К. В.* Обучение морфологического анализатора на большой электронной коллекции текстовых документов. Труды седьмой всероссийской научной конференции — Ярославль: Ярославский государственный университет, 2005. — С. 173–181.
9. *Сокирко А. В.* Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка // М.: 2005
10. *Черненьков Д. М.* Предсказание морфологических характеристик и парадигм словоизменения несловарных словоформ в текстах на русском языке // Киев: Труды конференции Мегалинг, 2009
11. *Alpaydin E.* Introduction to Machine Learning // The MIT Press 2004
12. *Friedman J., Hastie T., Tibshirani R.* The Elements of Statistical Learning: Data Mining, Inference, and Prediction // Springer, Stanford 2009
13. *Очагова Л. Н., Маслякова В. М., Зайцева Е. М., Дунаевская С. М.* Универсальная технология формирования словаря баз данных CDS/ISIS с использованием основ терминов // Государственная публичная научно-техническая библиотека России, М.: 2000
14. *Cao Z., Qin T., Liu T. Y., Tsai M. F., Li H.* Learning to rank: From pairwise approach to list wise approach // Proceedings of the 24th International Conference on Machine Learning. Corvallis, OR, 2007
15. *Xia F., Liu T. Y., Zhang J. W., Li H.* Listwise Approach to Learning to Rank — Theory and Algorithm // Proceedings of the 25th International Conference on Machine Learning. Corvallis, OR, 2008
16. *Banerjee A.* An Analysis of Logistic Models: Exponential Family Connections and Online Performance // Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX), 2007