DOI: 10.30888/2415-7538.2018-09-1-009

УДК 004.912

AUTOMATED DETECTION OF WORD MORPHOLOGICAL PARADIGM USING TEXT CORPORA

АВТОМАТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ МОРФОЛОГИЧЕСКОЙ ПАРАДИГМЫ СЛОВА ПО КОРПУСУ ТЕКСТОВ

Martirosyan A.A./Мартиросян А.А.
Московский государственный университет им. М.В.Ломоносова,
Москва, Ленинские Горы 1, стр. 52, 119234
Lomonosov Moscow State University
Moscow, Leninskie Gory 1, str. 52, 119234

В работе рассматривается проблема определения морфологической парадигмы слова. Предложенные в работе алгоритмы автоматического определения парадигмы слова используют тот факт, что неизвестное слово может быть использовано в корпусе текстов несколько раз в разных формах, что позволяет улучшить результаты предсказания. Обозначения парадигм используют систему обозначений морфологического анализатора RMU.

Ключевые слова: *морфологический анализатор, парадигма слова, корпус*

Вступление.

В настоящее время уделяется огромное внимание проблеме обработки текстов. Люди генерируют больше количество информации в сети Интернет. Эта информация может быть полезна для широкого спектра задач. При обработке больших объемов текста на естественном языке необходим морфологический анализатор. Морфологический анализатор нужен для того, чтобы определять грамматическую информацию слова. На примере анализатора pymorphy2 разберем разбор несловарных слов:

1) Если слово начинается с известного префикса, то он отсекается, разбирается то, что осталось и в конце префикс приписывается обратно.

2) Если слово начинается с неизвестного префикса, то программа попытается представить это слово, как <некий префикс> + какое-то слово из словаря. Если попытка удачная, то pymorphy2 считает, что данное слово разбирается так же, как и другое слово. Но есть ограничения: длина словарного слова должна быть не меньше 3, длина префикса не должна быть больше 5 и словарное слово - это существительное, прилагательное, глагол, причастие или деепричастие.

Если разбор по префиксу не дал результат, то есть возможность разбора по окончанию. Для того, чтобы предсказать формы слов по тому, как слова заканчиваются, при конвертации словарей pymorphy2 [1] собирает статистику по окончаниям: для каждого возможного окончания слова (от 1 до 5 букв) сохраняются все возможные разборы. Разбор сводится к поиску наиболее длинной правой части разбираемого слова. В данной работе предлагается метод определения парадигмы слова, основанный на разборе окончания и использования корпуса.

Основные определения

Лексема – набор всех форм одного слова. Например, дом, дома, домами находятся в одной лексеме.

Лемма (нормальная форма слова) – каноническая форма слова.

Псевдооснова – неизменяемая часть слова (в отличие от обычной основы включает приставку и не учитывает чередование).

Словоформа – форма некоторого слова русского языка.

Парадигма (модель изменения слова) – правила, по которым можно получить все словоформы в лексеме для заданной псевдоосновы.

Граммема – это морфологический описатель, относящий словоформу к какому-то морфологическому классу, например, словоформе “стол” с леммой СТОЛ будут приписаны следующие наборы граммем: “мр, ед, им, но”,“мр, ед, вн, но”.

Таким образом, морфологический анализ выдает два варианта анализа словоформы “стол” с леммой СТОЛ внутри одной морфологической интерпретации: с винительным(вн) и именительным(им) падежами. В результате морфологического анализа выдается информация о парадигме слова и грамматическая информация:

1. Часть речи
2. Одушевленность (для сущ.)
3. Род. Для некоторых слов существует понятие “общего рода”. Такие слова могут употребляться применительно к людям женского или мужского пола.
4. Число
5. Падеж
6. Возвратность (для глаголов)
7. Совершенность
8. Лицо глагола
9. Время
10. Залог

Парадигмы хранят в себе всевозможные словоформы и грамматическую информацию о них. Например, для слова “палка”:

Таблица 1

Словоформа	Псевдооснова	Суффикс
палка	пал	ка
палкам	пал	кам
палках	пал	ках
палке	пал	ке
палки	пал	ки
палкой	пал	кой
палку	пал	ку
палок	пал	ок

В нашем случае, парадигма – это суффиксы и грамматическая информация для каждой словоформы (для некоторых слов может быть префикс). Для оптимизации хранения, суффиксы и грамматическая информация записываются в отдельные массивы, а вместо них в таблице парадигмы выставляются

индексы.

Выделение окончаний

Большинство морфологических анализаторов хранят окончания для анализа несловарных слов. В этой работе используется такой же подход. Для создания таких словарей используются внутренние словари модуля RMU[3] и результаты анализа слов. Для каждого результата анализа находится номер парадигмы и псевдооснова слова. Далее, в словаре парадигм мы получаем окончания для данной парадигмы. Для каждого окончания мы берем несколько последних букв Псевдооснова и добавляем окончание. В итоге, для одного стема получается набор из “окончание псевдоосновы + окончание из парадигмы” (в дальнейшем будем называть эту конструкцию - «суффикс словоформы»). В модуле RMU существуют парадигмы с вложенными окончаниями. Например, у окончания “ящ” есть вложенная парадигма “AdjP1”. Это означает, что окончание “ящ” будет расширено всеми окончаниями из парадигмы “AdjP1”. В таких случаях записываются сразу расширенные версии окончаний. Для каждого суффикса словоформы записываются также различные парадигмы, которые образуют слова с данным суффиксом, и их количество. Раннее говорилось, что мы берем несколько окончаний псевдоосновы. В данной работе было принято решение поэкспериментировать с длиной выбора окончаний. Были выделены суффиксы словоформ с последними 2, 3 и 4 буквами псевдоосновы. В результате получилось 114930 различных суффиксов для словаря с 2 буквами, 311025 для словаря с 3 буквами и 643033 для словаря с 4 буквами.

Алгоритм разбора слова

Основная идея предложенного метода разбора слов - разбор с помощью дополнительных слов из корпуса. Для каждого входного слова мы пытаемся предсказать возможную парадигму по суффиксу. В случае удачного предсказания мы получаем парадигму либо набор парадигм. Для каждой предсказанной парадигмы генерируем все словоформы по данной парадигме. Делается это следующим образом: по парадигме получаем список всех флексий, по которому склоняются слова с этой парадигмой. По этим флексиям мы можем найти предполагаемую псевдооснову нашего слова. Имея псевдооснову слова, строим словоформы: <псевдооснова> + <окончание флексии>. Например, если псевдооснова - “самолет”, а одно из окончаний флексий - “ами”, то получится словоформа - “самолетами”. Таким образом получаются все словоформы данной леммы. Имеем набор словоформ для каждой предсказанной парадигмы. Введем ранжирование предсказанных парадигмы. Изначально все они имеют ранг 0. Берем набор словоформ, для каждого слова из него проверим, находится ли оно в исходном корпусе. Если находится совпадение, то увеличиваем ранг парадигмы на 1. Таким образом мы проверим все возможные парадигмы слова. Результирующей парадигмой будет та, у которой больше ранг. Если с парадигм с максимальным рангом несколько, то все будут выбраны, как верный результат. Если же мы не нашли никаких совпадений словоформ в корпусе и ранг остался нулевым, то считаем, что корпусной предсказатель в данном случае не помог.

Результаты эксперимента

В качестве эксперимента было решено смоделировать идеальный корпус - такой, в котором встречаются все словоформы анализируемого слова. Для этого из словаря RMU были выбраны случайно 1000 слов. Для каждого слова составляем полный список словоформ. Мы это можем сделать с помощью словаря парадигм и стема слова. Зная парадигму слова, мы можем сразу определить стем - смотрим, на какой суффикс оканчивается наше слово, из слова “вырезаем” этот суффикс. И уже когда у нас есть стем, то прибавляем к стему все оставшиеся суффиксы. Таким образом получаем слова со всеми своими словоформами. Например, для слова “самолет” получится такой список: самолет : самолета, самолетами, самолеты .... Для данного эксперимента был предложен иной алгоритм предсказания парадигмы. Для каждого слова и всех его словоформ мы выполняем предсказание только по суффиксу. То есть, используя словари, построенные на этапе выделения окончаний, мы пытаемся найти, на какой суффикс оканчивается наше слово и словоформы. Таких суффиксов может быть несколько. И каждое совпадение нам дает набор возможных парадигм. В итоге, для каждой леммы мы получаем несколько таких наборов(как минимум по одному набору на каждую словоформу). Основная идея - это составить пересечение множеств данных наборов. В этом пересечении останутся только парадигмы, которые есть во всех наборах. Разберем работу на примере:

Исходное слово - “самолет”, для облегчения примера мы допустим, что у этого слова одна словоформа, кроме леммы - “самолетами”. Мы делаем предсказание по суффиксу для исходного слова - пусть мы нашли 2 суффикса - “олет” и “ет”. У первого набор парадигм: (“Verb3_10”, “Verb3_6.1”, “Verb4_10”). У второго: (“Adj1_1”, “NounG_1”, “NounG_1.13”, “NounM_1”). Теперь предскажем для словоформы “самолетами” - он оканчивается только на суффикс “етами”. Данный суффикс содержит парадигмы (“NounG_1”, “NounG_1.13”). Выполняем пересечение этих трех наборов. Получим две возможные парадигмы - (“NounG_1”, “NounG_1.13”). Эти парадигмы и станут результатом предсказания. В реальности у леммы больше словоформ и пересечение их возможных словоформ дает более точный результат. Данный алгоритм был применен к случайному набору из 1000 слов, описанному ранее. В качестве правильного ответа использовалась выдача модуля RMU. Базовым методом для сравнения выберем алгоритм предсказания по суффиксу. Рассмотрим следующие метрики качества: точность (precision), полноту (recall), F-меру и долю правильных ответов (accuracy) — далее ДПО. Для расчета полноты и точности будем использовать микро-усреднение, так как у нас фактически небинарная классикация. В таблице ниже (single) - предсказание только по суффиксу.

Такой метод показывает хорошие результаты по сравнению с предсказанием по суффиксу. Но в чистом виде он неработоспособен, потому что во время анализа слова нам никто не будет давать словоформы этого же слова. Так же следует сказать, что результаты предсказания по суффиксу так высоки, потому что словари основаны из этих же слов.

Таблица 2

Результаты предсказания

Словарь	Точность	Полнота	F-мера	ДПО
2 буквы стема	0.13	0.77	0.23	0.31
3 буквы стема	0.35	0.83	0.49	0.48
4 буквы стема	0.57	0.86	0.68	0.60
2 буквы (single)	0.04	0.51	0.08	0.14
3 буквы (single)	0.16	0.76	0.26	0.31
4 буквы (single)	0.43	0.87	0.58	0.52

Следующий эксперимент - это корпусной предсказатель на совершенно незнакомых словах. Для этого из корпуса был выделен список слов, которые не встречались в словарях RMU. Таких слов получилось примерно 10930. После этого парадигма каждого слова из списка предсказывалась алгоритмом корпусного предсказателя. Сложность проверки результата предсказания заключается в том, что у нас нет правильных парадигм RMU для этих слов. Поэтому было принято решение провести проверку с помощью морфологического анализатора pymorphy2 и сопоставлять его выдачу с форматом парадигм RMU. Парадигмы из pymorphy2 и RMU считались одинаковыми, если совпадали следующие граммемы: часть речи, число, падеж, род, одушевленность, лицо, время, вид и категория переходности (если эти свойства есть у слова). В таблице ниже приведены количество верных результатов и количество предсказанных. Второе значение - это фактически количество слов, у которых были другие формы в списке. Также приведены отношение количества верных ответов к количеству предсказанных (T/P) и верных к общему количеству (T/A).

Таблица 3

Результаты

Словарь	Кол-во верных	Кол-во предсказанных	T/P	T/A
2 буквы основы	5310	6903	0.76	0.48
3 буквы основы	4821	6328	0.76	0.44
4 буквы основы	4189	5614	0.74	0.38

Из результатов видно, что количество предсказанных слов уменьшается с ростом количество букв псевдоосновы в словарях. Это связано с тем, что чем больше букв псевдоосновы мы берем, тем специфичнее получаются наши суффиксы. Но соотношение количества верных и предсказанных почти не изменяется.

В следующем эксперименте список слов был выбран следующим образом: выбирались те слова из корпуса, которые входят в словари RMU. Это делается для того, чтобы не возникало проблем с проверкой результатов (как в прошлом эксперименте). Мы получаем “реальное” количество словоформ для леммы в корпусе и в то же время можем проверить результаты. Таких слов было примерно 110 тысяч. Список был уменьшен до 1000 слов. В таблице приведены

результаты предсказания с помощью алгоритма корпусного предсказателя. Эти же слова были проанализированы с помощью алгоритма суффиксного предсказания, эти результаты указаны в таблице (single).

Таблица 4

Результаты

Словарь	Точность	Полнота	F-мера	Accuracy
2 буквы основы	0.15	0.37	0.22	0.24
3 буквы основы	0.37	0.55	0.44	0.43
4 буквы основы	0.53	0.64	0.56	0.56
2 буквы (single)	0.001	0.90	0.002	0.009
3 буквы (single)	0.02	0.99	0.05	0.10
4 буквы (single)	0.16	0.99	0.27	0.31

В случае предсказания только по суффиксу были получены довольно слабые результаты, только с 4 буквами стема можно получить удовлетворительные результаты, когда как в случае алгоритма поиска по корпусу даже для 3 букв результаты лучше.

Был проведен еще один эксперимент. В качестве базового метода был выбран алгоритм корпусного предсказателя, но на этапе выбора результирующего ранга выбирались 2 ранга - максимальный и следующий по рангу. Приведены результаты для этого эксперимента:

Таблица 5

Результаты

Словарь	Точность	Полнота	F-мера	Accuracy
2 буквы стема	0.04	0.25	0.08	0.12
3 буквы стема	0.15	0.38	0.21	0.22
4 буквы стема	0.36	0.54	0.43	0.42

Как видно из результатов, данный эксперимент не дал положительных результатов.

Литература:

1. M. Korobov, “Morphological analyzer and generator for Russian and Ukrainian languages,” in Analysis of Images, Social Networks and Texts, Communications in Computer and Information Science, 2015.

2. Волкова И.А. Адаптация и обучение системы общения с ЭВМ на естественном языке. Электронный ресурс” http://axofiber.org.ru/ download/volkova-dissertation.pdf. Дата последнего обращения: 10.03.2018

3. Волкова И.А., Проскурня М.О. Программный комплекс для лингвистической обработки текстов на русском языке. В сборнике

«Труды Международного семинара по компьютерной лингвистике и ее приложениям Диалог 2002»: М., Наука, том 1, с. 96-99.

4. Грановский Д.В., Бочаров В.В., Бичинева С.В. Открытый корпус: принципы работы и перспективы // Компьютерная лингвистика и развитие семантического поиска в Интернете: Труды научного семинара XIII

Всероссийской объединенной конференции «Интернет и современное общество». Санкт-Петербург, 19–22 октября 2010 г. / Под ред. В.Ш. Рубашкина. — СПб., 2010. — 94 с.

Abstract.

The article covers the problem of determining the morphological paradigm of a word. In this paper the algorithms for automatic definition of word paradigm are presented. The principle of their work is based on the fact that an unknown word can be used in the text corpora several times in different forms. This improves the prediction results. Paradigm designations use the notation of the RMU morphological analyzer.

Key words: *morphological analyzer, word paradigm, NLP, text corpora*

References:

1. M. Korobov, “Morphological analyzer and generator for Russian and Ukrainian languages” in Analysis of Images, Social Networks and Texts, Communications in Computer and Information Science, 2015.

2. I. Volkova, “Adaptation and training of communication system with computer in natural language” http://axofiber.no-ip.org/ download/volkova-dissertation.pdf.

3. I.Volkova, M.Proskurnya, “Programs for linguistic processing of Russian texts” in Computational Linguistics and Intellectual Technologies. Proceedings of the conference. Protvino, 2002, pp. 96-99

4. Bocharov V., Bichineva S., Granovsky D., Open corpora: operating principles and prospects // Computational linguistics and development of semantic search on the Internet: Proceedings of the scientific seminar of the XIII all-Russian United conference "Internet and Modern Society". Saint Petersburg, October 19-22, 2010 / edited by V. S. Rubashkin. — SPb., 2010. - 94 p.

Научный руководитель: *доцент, к.ф.-м.н.* Головин И.Г.

Статья отправлена: 10.03.2018 г.