Особенности построения морфопроцессора русского языка CrossMorhpy

Сапин А.С., МГУ имени М.В. Ломоносова, факультет ВМК
alesapin@gmail.com
Большакова Е.И., МГУ имени М.В. Ломоносова, факультет ВМК,
НИУ Высшая школа экономики
eibolshakova@gmail.com

Аннотация

Характеризуется новый морфологический процессор с открытым кодом, разработанный для русского языка и основанный на словарной морфологии. Функциональные особенности процессора рассматриваются в сравнении с наиболее часто используемыми свободно доступными морфологическими парсерами. Описываются принципы построения и архитектура разработанного процессора, а также применяемые методы обработки несловарных словоформ, разрешения морфологической омонимии и морфемного разбора слов.

1 Введение

В настоящее время компьютерная лингвистика становится все более популярным направлением компьютерных наук. Причиной является активный рост объемов хранимой текстовой информации в электронном виде и необходимость ее автоматической обработки.

Начальные этапы анализа текста на естественном языке обеспечивают графематические и морфологические процессоры. Модули графематического анализа работают на уровне потока входных символов, обеспечивая выделение лексических или структурных единиц: словоформ, знаков препинания и т.п. Морфологические модули (парсеры) выполняют функции, связанные с морфологическим анализом выделенных словоформ. Основной задачей морфологического анализа является определение по заданной словоформе леммы (нормальной формы слова) или основы слова, а при необходимости, и набора характеризующих ее морфологических характеристик (тегов), к числу которых относится род, число, одушевленность, падеж, лицо и др.

Морфологический анализ является одной из самых разработанных областей компьютерной лингвистики. Для русского языка она имеет ряд особенностей, что связано с его высокой флективностью и наличием большого числа исключений. К особенностям можно отнести наличие 9 падежей у существительных, кратких форм у прилагательных, отсутствие форм настоящего времени у глаголов совершенного вида и др. Все это значительно усложняет создание морфологических процессоров русского языка. Одна из существенных проблем – омонимия, когда у одной словоформы (к примеру: *стекла*, *плачу*) может оказаться несколько различных лемм и/или наборов морфологических характеристик.

В настоящий момент известно более десятка работоспособных морфологических парсеров русского языка, однако все они различаются набором функциональных свойств и используемых морфологических тегов, а также качеством работы – см., например, [Kuzmenko, 2016]. Важной особенностью является также доступность парсера и открытость его кода, что позволяет применять его для различных исследований и построения программных приложений.

В данной работе характеризуется новый морфологический процессор CrossMorphy с открытым кодом[1], реализованный на языке C++ с использованием свободно доступных словарных данных Open Corpora[2]. Его разработка была инициирована развитием проекта LSPL[3], в рамках которого создана инструментальная система для построения различных приложений для обработки русскоязычных текстов на базе поверхностного синтаксического анализа с использованием лексико-синтаксических шаблонов [Большакова, 2015; 2016]. Для построения на базе этой системы более сложных приложений требуется морфологический процессор с определенной функциональностью, и разработка процессора CrossMorphy призвана удовлетворить эту потребность.

[1] https://github.com/alesapin/XMorphy
[2] http://opencorpora.org
[3] www.lspl.ru

Описание функциональных особенностей процессора CrossMorphy предваряется обзором основных возможностей и характеристик морфологических парсеров для русского языка на примере наиболее известных свободно доступных модулей. К числу важных возможностей CrossMorphy относится, кроме полного морфологического анализа словоформ, контекстное разрешение (снятие) морфологической омонимии, предсказание морфологических характеристик неизвестных (несловарных) словоформ, а также определение морфемного состава слов.

2 Морфологические парсеры для русского языка

В данном разделе рассматриваются наиболее популярные и доступные для открытого использования морфологические парсеры русского языка. Большинство из них: aot[1], mystem[2], pymorphy[3] были участниками соревнования по морфологическому анализу MorphoRuEval-2010 [Ляшевская, 2010] и нередко используются в практических приложениях. Их реализации базируются на классическом методе словарной морфологии и так или иначе восходят к русскому грамматическому словарю А.А. Зализняка.

Морфопроцессоры анализируются с точки зрения основных функциональных возможностей: *лемматизация* (получение нормальной формы слова), *стемминг* (получение основы), *полный морфологический анализ* (получение кроме леммы набора всех значимых морфологических характеристик), *морфологический синтез* (генерация нужной словоформы слова по ее морфологическим характеристикам, или синтез всей его парадигмы слова), *предсказание* (классификация) новых слов, *разрешение* (*снятие*) морфологической омонимии. Кроме этого, рассматриваются некоторые их технологические характеристики, такие как размер словаря, возможности его расширения или дополнения, открытость исходных кодов.

Морфологический процессор aot [Сокирко, 2004] – исторически первый открытый модуль на базе словаря А.А.Зализняка (более 161 тыс. лемм), предоставляющий все функции полного морфологического анализа, а также синтеза словоформ. Он до сих пор встроен в инструментальную систему на базе языка LSPL [Большакова, 2016]. Однако процессор не имеет удобных средств для пополнения словаря и в настоящее время не поддерживается.

Мофологический парсер mystem базируется на словаре НКРЯ[4] (более 250 тыс. лемм) и является наиболее мощным и стабильным по функции полного анализа, а также снятия морфологической омонимии. Исходные коды mystem являются закрытыми, однако парсер позволяет подключать собственные словари (через опции командной строки или интерфейса библиотеки). В этом случае стандартный словарь полностью заменяется пользовательским.

Сравнительно недавно разработанный, но уже широко применяемый процессор pymorphy2 [Korobov, 2015] – модуль с открытым исходным кодом, предоставляющий все функции полного морфологического анализа, включая синтез и разрешение омонимии (последняя – бесконтекстным методом). Процессор использует словарные данные проекта OpenCorpora (около 350 тыс. лемм), которые регулярно пополняются.

Кроме этих трех морфологических процессоров для русского языка наибольший интерес представляет парсер TreeTagger[5]. Изначально проект TreeTagger[6] позиционировался как система для определения частей речи с возможностью настройки на любой естественный язык при наличии морфологически размеченного корпуса текстов. Проект поддерживается, создаются новые словари и парсеры под различные языки. Процессор для русского языка позволяет проводить полный морфологический анализ и распространяется только в виде бинарного файла. Синтез словоформ в TreeTagger отсутствует, как и возможность подключения дополнительных словарей.

В Таблицах 1 и 2 собраны данные соответственно о функциональных и технологических характеристиках рассмотренных морфологических парсеров. В последнем столбце указаны характеристики разработанного процессора CrossMorphy.

Как видно из Таблицы 1, все процессоры предоставляют наиболее важные для русского языка функции лемматизации и полного

[1] http://aot.ru/docs/rusmorph.html
[2] https://tech.yandex.ru/mystem/doc/
[3] http://pymorphy2.readthedocs.io/en/latest/index.html
[4] http://ruscorpora.ru
[5] http://corpus.leeds.ac.uk/mocky/
[6] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger

морфологического анализа, и большинство процессоров – функцию разрешения омонимии. Везде реализована возможность предсказания характеристик (классификации) новых, несловарных слов. Функция стемминга реализована не везде, т.к. менее востребована на практике, однако все процессоры, кроме TreeTagger, предоставляют возможность получения словоизменительной парадигмы заданной словоформы, а с ее помощью достаточно просто получить псевдооснову слова. Морфологический синтез также реализован лишь в двух из рассмотренных процессоров, хотя во многих задачах компьютерной лингвистики данная функция является важной.

Что касается технологических особенностей (Таблица 2), то два из представленных парсеров являются закрытыми и распространяются исключительно в виде бинарных файлов. Словарь парсера mystem является закрытым, словарь TreeTagger доступен только в виде бинарного файла. Скорость обрабатываемых слов у всех процессоров является достаточно высокой, объем словаря представителен у большинства. Возможность подключения словаря является особенной важной для задач из ограниченных предметных областей. Данную функцию предоставляет только mystem.

Таблица 1. Функциональные характеристики морфологических парсеров

Парсер	aot	mystem	TreeTagger	pymorphy2	CrossMorphy
Лемматизация	+	+	+	+	+
Стемминг	–	–	–	–	+
Полный морфолог. анализ	+	+	+	+	+
Предсказание характеристик	+	+	+	+	+
Снятие омонимии	–	+	+	+	+
Синтез словоформ	+	–	–	+	+

Таблица 2. Технологические характеристики морфологических парсеров

Парсер	aot	mystem	TreeTagger	pymorphy2	CrossMorphy
Открытость исходного кода	+	–	–	+	+
Открытость словаря	+	–	–	+	+
Подключение словарей	–	+	–	–	–
Объем словаря (тыс. лемм)	160	> 250	210	390	390
Число разборов в сек. (тыс.)	60-90	100-120	20-25	80-100	~ 28

Проведенное сравнение показывает специфику функциональных свойств разработанного процессора CrossMorphy, для проекта LSPL необходима вся указанная в Таблице 1 функциональность, а также открытость кода и словаря. Дополнительно, для нужд проекта в CrossMorphy реализован морфемный разбор слов (т.е. разбиение на составляющие морфы, например: *раз-бег-ающ-ий-ся*), расширяющий возможности морфологического процессора, в частности, с его помощью можно более точно классифицировать новые слова с уже известными корнями.

Из рассморенных модулей наиболее подходящим для проекта был бы pymorphy2, однако он реализован на интерпретируемом языке python, что исключает возможность построения автономного приложения.

3 Принципы построения процессора CrossMorhpy

CrossMorphy представляет собой кроссплатформенную библиотеку и утилиту командной строки, написанные на языке C++.

С архитектурной точки зрения, библиотека включает 4 больших компоненты. Первой компонентой является графематический анализатор для разбиения потока входных символов на 7 классов токенов (словоформы в кириллице, знаки препинания, разделители, числа, буквенно-цифровые комплексы, иероглифы). Как показывает наш опыт использования морфологических парсеров, наличие собственного модуля сегментации на токены упрощает в ряде аспектов реализацию (и применение) морфологического анализа.

Второй компонентой является собственно морфологический процессор, реализующий

функции морфологического разбора и генерации (синтеза) словоформ. Компонента представляет собой иерархическую структуру классов-обработчиков, каждый из которых добавляет новую функциональность, используя методы своего предка. Обработчики отвечают соответственно за морфологический разбор словоформы по встроенному словарю и предсказания характеристик несловарной словоформы по одному из эвристических методов. Методы учитывают соответственно префиксы, окончания, дефисы словоформы. Базовый обработчик включает возможности всех остальных.

Обработка словоформы происходит по цепочке: сначала разбор по словарю, в случае неудачи по префиксу, а затем по окончанию. Разбор слов с дефисом происходит отдельно, при этом для составных частей слова с дефисом используются другие обработчики.

Третью компоненту образуют модули разрешения омонимии, бесконтекстным и контекстным методом. Они работают с набором разборов-омонимов, полученных от базового обработчика словоформы, выдавая на выходе один, наиболее вероятный разбор. Здесь и далее разбором мы называем совокупность леммы и всех значимых морфологических характеристик исходной словоформы.

К четвертой компоненте относятся вспомогательные модули, поддерживающие преобразования между различными версиями морфологических тегов, а также модуль для обучения классификаторов, разрешающих омонимию.

Ключевыми решениями при разработке CrossMorphy был выбор модели морфологии и соответствующих словарных данных. Для высокофлективного языка (каковым является русский) традиционным подходом к решению задачи морфологического анализа является модель на основе словаря словоформ. При наличии качественного и объемного словаря это дает хорошие характеристики точности и полноты морфологического разбора.

Соответственно, для разработки процессора были выбраны обширные и свободно доступные словарные данные системы Open Corpora [Bocharov, 2011], на основе которых был построен внутренний словарь словоформ в виде DAFSA-автомата (deterministic acyclic finite state automaton) [Daciuk, 2000].

Словоформы в таком автомате выступают ключами. При этом словоформы с одинаковыми префиксами хранятся вместе, что позволяет существенно экономить необходимую память. Значения, являющиеся морфологическими характеристиками, находятся сразу после словоформ, как правило за символом-разделителем, который не может встретиться ни в одной из входных словоформ. Поиск словоформы в таком автомате происходит за время, линейное от длины входной словоформы: достаточно просто пройти все состояния автомата, которые соответствуют символам входной словоформы, и за разделительным символом получить все характеристики словоформы.

Отметим, что словари словоформ русского языка, как правило, содержат несколько миллионов единиц, что является достаточно большим объемом данных, который может не поместиться в оперативной памяти в процессе построения автомата. Однако, как показано в [Daciuk, 2000], такой автомат может быть построен "на лету".

Аналогичные структуры автомата применяются в парсерах aot и pymorphy2 (исходные коды mystem являются закрытыми, и использованной структуры данных не известны).

Основным отличием CrossMorphy от процессоров pymorphy2 и aot в реализации словаря является формат его хранения. DAWG-автомат используется только для хранения индекса словоформ, а данные о морфологических характеристиках и парадигмах закодированы в виде монолитного массива в отдельном участке памяти. Данный подход является менее компактным, чем в pymorphy2: словарь занимает примерно в 1,5 раза больше памяти, но обеспечивает, предположительно, большую скорость доступа к элементам.

Достоинством словарных морфологий являются возможность выполнения как анализа, так и синтеза словоформ, исходя из леммы или другой словоформы, а также заданного набора морфологических характеристик нужной словоформы.

В процессоре aot реализован только синтез всей парадигмы по лемме. pymorphy2 по входной словоформе сначала определяет возможные ее разборы, а потом для одного или нескольких разборов генерирует нужную словоформу. В CrossMorphy реализована подобная функциональность, но также возможен синтез нужных словоформ исходя из конкретной словоформы и заданных морфологических характеристик, причем если морфологические характеристики указаны не полностью, например, для существительного

задано только число, то выводятся все допустимые формы (для *шарами* будет получено *шар*, *шара*, *шару*, *шаром*, *шаре*)

Слабым местом словарных морфологий является анализ несловарных (неизвестных) словоформ, а также необходимость разрешения морфологической омонимии. При разработке CrossMorphy этим проблемам было уделено существенное внимание.

4 Обработка несловарных слов

Если обрабатываемая словоформа не была найдена в словаре, то в этом случае процессор последовательно пытается предсказать ее морфологические характеристики и лемму на основе эвристических методов: по префиксу словоформы или по ее окончанию. Неизвестные слова с дефисом также обрабатываются по специальным правилам.

Схожие методы используются в парсерах aot, pymorphy2, TreeTagger, но по каждому имеются существенные отличия. В частности, в aot в отличие от CrossMorphy слова с дефисами всегда разбираются как единое целое (тем самым, не классифицируются словоформы вида *S-выражение*). TreeTagger в отличие от CrossMorphy предсказывает по окончанию только один вариант разбора словоформы.

Метод предсказания по окончанию основан на известном принципе аналогии, согласно которому одни и те же окончания слов с большой вероятностью соответствуют одному синтаксическому (словоизменительному) классу с конкретными морфологическими характеристиками. Для их определения в CrossMorphy используется заранее подсчитанная (по заданному корпусу) статистика встречаемости всех возможных окончаний слов (от 1 до 5 букв), по которой для каждого окончания определяется самая частотная часть речи и возможные для нее при данном окончании наборы морфохарактеристик.

Первым шагом предсказания является попытка найти существующую в словаре словоформу языка, которая имела бы максимально общее окончание со входным словом. Затем по этому общему окончанию берется наиболее частотная часть речи и соответствующие возможные наборы морфохарактеристик. Для этого используется дополнительный DAWG-автомат, в котором окончаниям с определенными вероятностями соответствуют наборы морфологических характеристик. По совпавшему окончанию выбирается наиболее вероятные для него морфологические характеристики.

Таким образом, отличием реализованного метода предсказания по окончанию является сохранение всех возможных вариантов разбора для самой частотной части речи.

Существенное отличие реализованного в CrossMorphy метода предсказания по префиксу (от aot и pymorphy2) заключается в том, что не проводится анализ по неизвестному префиксу. Применяется только отсечение известного префикса (из встроенного списка префиксов) с последующим поиском остатка в основном словаре системы, что позволяет избежать неверных вариантов разбора.

Указанные отличия сказываются, например, в том, что для существительного *вейпер* pymorphy2 среди прочих выдает заведомо ложный вариант глагольной формы и в отличие от CrossMorphy не справляется с анализом слова *авторша*.

Что касается слов с дефисом, то многие из них уже есть в словаре CrossMorphy (*Комсомольск-на-Амуре*, *баба-яга*). Для несловарных словоформ последовательно применяются следующие правила.

1) если в слове более одного дефиса (например, *фолк-панк-рок*), то оно разбирается по методу предсказания по окончанию;

2) если в слове одна из частей (первая или вторая) является числом или известным префиксоидом (*Маяк-401*, *вице-директор* и т.д.), то происходит разбор только оставшейся части, которая дает набор морфохарактеристик и лемму, а результирующая лемма получается приписыванием отсеченной части;

3) если обе части слова через дефис являются одинаковыми (*гули-гули*, *тук-тук*), то происходит разбор только одной части;

4) если для первой или второй части слова с дефисом парсер не смог найти по словарю ни одного варианта разбора, то по отдельности обрабатываются следующие случаи:

а) если эта часть – слово в латинице или буквенно-цифровой комплекс (*α-конверсия, ER-метод, 3D-система*), то разбор делается для оставшейся части;

б) если среди вариантов разбора первой части слова встречается вариант, в котором нормальная форма совпадает с исходной словоформой (*киловатт-часов, веб-дизайн*, *лексико-семантический*), то разбор делается для второй части слова;

г) в противном случае (*человек-гора*, *избачитальня*) обе части слова с дефисом анали-

зируются независимо и ко всем вариантам разбора второй части приписывается результат разбора первой части, часть речи которого совпадает с частью речи второй части.

Рассмотренные правила позволяют обрабатывать словоформы из открытого класса слов с дефисом.

5 Разрешение морфологической омонимии

Хорошо изученной для многих естественных языков задачей является разрешение (снятие) частеречной омонимии, точность ее решения достигает 98-99% – как для статистических методов, так и методов, основанных на правилах. Однако при наличии размеченных текстовых корпусов чаще применяются статистические методы.

В частности, для снятия частеречной омонимии в парсере TreeTagger[1] используются решающие деревья. В узлах такого дерева находятся предикаты с ответом "да" или "нет" для двух предшествующих слов. При этом в листьях хранятся значения вероятностей для возможных ответов. Построение дерева происходит рекурсивно, с помощью модифицированного алгоритма ID3. На каждом шаге, для двух предыдущих слов проверяются все возможные предикаты на равенство всем частям речи, при этом для определения лучшего используется правило максимизации энтропии Шеннона. Для определения части речи входного слова достаточно, используя информацию о предыдущих словах, пройти по дереву от корня до листьев и выбрать наиболее вероятное значение.

В последнее время с появлением размеченных корпусов для русского языка для разрешения омонимии применяются в основном статистические методы. В одной из первых работ по снятию частеречной омонимии статистическим методом [Зеленков, 2005] достигалось 97,42% точности.

Для всех высокофлективных языков, в том числе для русского, более важна задача полного разрешения омонимии, когда однозначно определяется не только часть речи и лемма обрабатываемой словоформы, но и весь набор значимых морфологических характеристик. В работе [Сокирко, 2005] были начаты исследования этой задачи, и в настоящий момент различные решения на основе статистических методов реализованы во многих парсерах.

Парсер MyStem может разрешать омонимию двумя способами: с учетом контекста и без учета. Снятие омонимии без учета контекста происходит благодаря обучению наивного баесовского классификатора на размеченном корпусе со снятой омонимией. Частоты встречаемости факторизируются и отдельно настраиваются для окончаний парадигм, основ парадигм и самих парадигм. Для контекстного снятия омонимии используется технология MatrixNet[2], основной идеей является ранжирование разборов на основе слов, ближайших к разбираемому слову.

В процессоре pymorphy2 реализовано только бесконтекстное снятие морфомонимии на основе статистики словоупотреблений в размеченном текстовом корпусе Open Corpora и принципе максимального правдоподобия. [Korobov, 2015]. Если слово имеет несколько вариантов разбора, то среди всех выбирается наиболее вероятный.

В CrossMorphy реализовано два метода снятия морфологической омонимии: контекстный и бесконтекстный. Снятие омонимии без контекста устроено проще, чем в pymorphy2, но также на основе корпусной статистики. Для каждого слова обучающего корпуса подсчитывается, сколько раз оно встретилось с определенным набором морфологических тегов. Результат сохраняется в DAFSA-автомате в виде четверки *(слово, часть речи, набор тегов, частота)*. Затем по всем вариантам разбора подсчитается суммарная встречаемость слова в корпусе с любыми морфологическими тегами и делится на частоту встречаемости с соответствующим разбором.

При применении этой статистики в ходе бесконтекстного разрешения омонимии словоформы допускается не полное (минимум 4 тега, не считая части речи) совпадение тегов словоформы с набором, найденным в словаре, поскольку примененный для обучения корпус не может содержать всех слов словаря.

Метод бесконтекстного разрешения морфологической омонимии рассматривается как вспомогательный для метода разрешения с учетом контекста. Последний реализован в CrossMorphy на базе метода условных слу-

[1] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf

[2] http://www.dataversity.net/yandex-boosts-precision-ad-targeting-machine-learning-method-matrixnet-is-behind-the-scenes/

чайных полей (CRF). Этот метод машинного обучения выбран как один из наиболее перспективных для данной задачи [Антонова, 2013; Muzychka, 2014].

Для контекстного снятия омонимии используется метод CRF версии lbgfs с регуляризацией L2. Снятие омонимии по всем морфологическим характеристикам (тегам) является сложной для обучения CRF-классификатора из-за большого числа тегов, что потребовало усложнения модели. Применяются четыре обученных CRF-классификатора, последовательно отсекающих омонимичные варианты. Первым работает CRF-классификатор для части речи, используемые им признаки – словоформа и возможные части речи (в виде бинарного вектора). Затем применяется классификатор для рода (признаки: словоформа, уже определенная часть речи, возможные варианты рода: мужской, женский, средний). После этого аналогичным образом работают CRF-классификаторы числа и падежа. Если после применения всех классификаторов по прежнему остается несколько вариантов разбора слова (например, омонимия по одушевленности), то используется бесконтекстное снятие. В итоге остается единственный вариант разбора.

Точность описанной модели была оценена по корпусу НКРЯ со снятой омонимией (~ 1 млн. словоформ). Оценки, получаемые после применения очередного классификатора, приведены в Таблице 3. Заметим, что разрешение омонимии в другой последовательности морфологических характеристик, чем указанная (род-число-падеж), не дало в экспериментах улучшения приведенных показателей.

Таким образом, точность реализованной модели достаточно высока и вполне соответствует оценкам из указанных выше работ, а также работе [Ляшевская, 2010], где показано, что значения такой классической характеристики, как F-мера определения леммы и морфологических характеристик для современных морфоанализаторов составляет от 85 до 98 процентов в зависимости от выборки анализируемых слов и контекстов.

Таблица 3. Точность CRF-классификаторов

CRF	Ч. речи	Род	Число	Падеж
Точность	97,94	97,03	96,61	93,42

6 Морфемный разбор

Современные морфологические процессоры рассматривают словоформу на уровне основы и окончания (чаще псевдоосновы и псевдоокончания). Однако, любой способ словообразования происходит на уровне *морфов* – минимальных значащих единиц языка, которые в русском языке делятся на корневые (корень слова) и служебные: префикс (приставка), суффикс, флексия (окончание), постфикс. Носителем основного смысла слова является корень, а служебные, в общем случае, придают дополнительный смысл.

Морфемный разбор слов расширяет возможности морфологического процессора, т.к. на его основе можно определять семантически близкие слова разных частей речи (из одного словообразовательного гнезда) и даже учитывать степень их близости по количеству различающихся аффиксов (например: *за-работ-анн-ый* и *за-работ-ать*).

Задача автоматического морфемного разбора слов является достаточно трудной и слабо изучена в компьютерной лингвистике. К основным сложностям относится: отсутствие однозначных (разделяемых всеми лингвистами) правил разбиения слова на морфы, наличие большого числа исключений, недостаточная полнота словарей, содержащих сведения о морфемном разборе слов.

К наиболее значимым результатам по морфемному разбору относятся работы [Harris, 1970] и [Bernhard, 2007], предлагающие методы, которые опираются на статистику из неразмеченных текстовых корпусов. Метод Харриса при выделении морфов слов достигал точности 89% , однако при существенном ограничении: он не позволял выделять морфы из одной буквы в начале и конце слова.

Для CrossMorphy задача морфемного разбора решалась как задача классификации на основе машинного обучения методом CRF на основе размеченных данных из словарей морфемного разбора системы КроссЛексика [Большаков, 2013] и Вики-словаря[1] (слова, входящие в оба источника, нередко разбивались на морфы по-разному). Объем обучаемых данных соответственно был 23,4 и 94,4 тыс. слов, разобранных на морфемы. Построенные классификаторы дали соответственно точность в 79% для данных КроссЛексики и

[1] https://ru.wiktionary.org/wiki/

69% для данных из Вики-словаря, что можно считать довольно точным результатом для рассматриваемой задачи.

7 Заключение

Разработанный морфологический процессор с открытым кодом CrossMorphy демонстрирует функциональность, которая может быть востребована при проведении различных исследований в области компьютерной лингвистики и построении приложений. Предполагается его интеграция в инструментальную систему на базе языка LSPL. Ближайшими задачами является всестороннее тестирование описанных функций процессора, а также оптимизация процедур поиска по словарю и обработки несловарных слов, чтобы приблизиться по скорости к показателям парсера pymorphy2.

Список литературы

Kuzmenko E. 2016. Morphological Analysis for Russian: Integration and Comparison of Taggers. // Analysis of Images, Social Networks and Texts. Fifth Int. Conference AIST 2016. Communications in Computer and Information Science, Vol. 661. Springer Berlin Heidelberg.

Большакова Е.И. 2014. Язык лексико-синтаксических шаблонов LSPL: опыт использования и пути развития // Программные системы и инструменты: Тематический сборник, №15 – М.: МАКС Пресс.

Большакова Е.И., Иванов К.М., Сапин А.С., Шариков Г.Ф. 2016. Система для извлечения информации из текстов на базе лексико-синтаксических шаблонов // Пятнадцатая национ. конференция по искусственному интеллекту с междунар. участием (КИИ-2016): Тр. конференции. Т. 1 – Смоленск, Универсум, 2016.

Ляшевская О.Н. и др. 2010. Оценка методов автоматического анализа текста: морфологические парсеры русского языка //Компьютерная лингвистика и интеллект. технологии: По материалам ежегодной Международной конференциии "Диалог". Вып. 9 (16) – М.: Изд-во РГГУ.

Сокирко А. В. 2004. Морфологические модули на сайте www.aot.ru // Компьютерная лингвистика и интеллект. технологии: Труды международной конференции Диалог'2004 – М.: Наука.

Korobov M. 2015. Morphological analyzer and generator for Russian and Ukrainian languages // Analysis of Images, Social Networks and Texts. Fourth Int. Conference AIST 2015. Communications in Computer and Information Science, Vol. 542. Springer Berlin Heidelberg.

Bocharov V., Bichineva S., Granovsky D., Ostapuk N., Stepanova M. 2011. Quality assurance tools in the OpenCorpora project // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Междунар. конференции «Диалог». Вып. 10 (17). – М.: Изд-во РГГУ.

Daciuk J. et al. 2000. Incremental construction of minimal acyclic finite-state automata //Computational linguistics, V. 26, №. 1.

Зеленков Ю.Г., Сегалович И.В., Титов В.А. 2005. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2005 – М.: Наука.

Сокирко А. В., Толдова С. Ю. 2005. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп) // Интернет-математика 2005 – М.: Яндекс, 2005.

Антонова А.Ю., Соловьев А.Н. 2013. Использование метода условных случайных полей для обработки текстов на русском языке // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной международной конференции "Диалог". Вып. 12 (19): в 2 т. Т.1 – М.: Изд-во РГГУ.

Muzychka S.A., Romanenko A.A., Piontkovskaja I.I. 2014. Conditional Random Field for Morphological Disambiguation in Russian //Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialouge" – M.: RGGU.

Bernhard D. 2007. Simple morpheme labelling in unsupervised morpheme analysis //Workshop of the Cross-Language Evaluation Forum for European Languages. Springer Berlin Heidelberg, 2007.

Большаков И.А. 2013. КроссЛексика - Универсум связей между русскими словами. // Бизнес-информатика, № 3, 2013.

Harris Z.S. 1970. Morpheme boundaries within words: Report on a computer test, Transformation and Discourse Analysis Papers 73, 1970.