Исследование инструментов морфологического анализа текстов на русском языке для повышения точности алгоритмов обработки в библиотеке JMorfSdk

А. Н. Рыкунов, email: casi.05@mail.ru [1]
Е. В. Полицына, email: kathrin.beaver@mail.ru [1]
С. А. Полицын, email: pul_forever@mail.ru [1]
А. С. Поречный, email: alex.porechny@mail.ru [1]

[1] Московский авиационный институт (национальный исследовательский университет)

Аннотация. *Стремительное развитие технологий приводит к тому, что с каждым годом не только появляется все больше различных инструментов обработки текста на естественном языке, но и развиваются существующие. В статье приводятся результаты сравнительного анализа различных инструментов морфологического этапа анализа текстов на русском языке. На основе полученных результатов определяются направления развития текущей версии библиотеки JMorfSdk, которая входит в состав фреймворка TAWT, и предлагается несколько путей по повышению качества и точности результатов производимого морфологического анализа.*

Ключевые слова: *морфологический этап анализа, обработка естественного языка, компьютерная лингвистика, русский язык, сравнительный анализ.*

Введение

Постоянное увеличение количества производимой текстовой информации приводит к необходимости разработки и развития инструментов автоматического анализа текста. Сложность естественного языка, а также грамматические особенности каждого из них требуют не только реализации инструментов обработки языка для каждого из них отдельно, но и разделения процесса анализа на несколько этапов.

Все более активно инструменты автоматического анализа текста стали использоваться в различных сферах применения прикладных информационных систем. Необходим постоянный анализ текстовой информации, поступающей от пользователей или имеющейся в системах. Поэтому важно, как увеличение точности обработки, так и

© Рыкунов А.Н., Полицына Е.В., Полицын С.А., Поречный А.С., 2021

эффективность использования памяти и скорости работы инструментов лингвистического анализа текстов.

Морфологический этап анализа текста предоставляет обобщение данных, полученные части речи которых в дальнейшем используются для построения синтаксических отношений между словами на уровне синтаксического анализа [1]. Кроме того, использование морфологических анализаторов даёт возможность увеличить полноту и точность информационного поиска [2].

На сегодняшний день инструменты морфологического анализа текста базируются на словарях, на вероятностных подходах, на конечных автоматах, машинном обучении и т.д. Такое разнообразие может обуславливаться тем, что даже в рамках морфологического анализа могут решаться различные задачи.

1. Библиотека JMorfSdk в составе фреймворка TAWT

Фреймворк TAWT (Tools for Automated Work with Text) включает в себя набор программных инструментов лингвистического анализа текстов на русском языке [6]. Все инструменты используют общую схему подключения, стандартную для платформы Java: используется глобальный репозиторий бинарных зависимостей, исходный код которых, примеры и ссылки на артефакты находятся в общем доступе на Github [5].

Инструмент JMorfSdk (Java Morphological Sdk) реализует морфологический этап анализа теста, основан на модификации машиноориентированного грамматического словаря русского языка А.А. Зализняка, разработанного и поддерживаемого проектом OpenCorpora [4], используется версия 2.10.17. Корпус содержит на данный момент 360 тысяч уникальных слов, 5 млн. словоформ, которые включают редкие и новые слова, а также самые типичные опечатки для уменьшения их влияния на анализ текста в целом.

Алгоритмы анализа, реализованные в библиотеке JMorfSdk, имеют высокую производительность за счет использования хэш-таблиц вместе с использованием битовых операций и хранением самых необходимых характеристик в битовой шакале, что позволило получить константную сложность определения множества омоформ слова и их морфологических характеристик. Средняя скорость выполнения морфологического анализа текста составляет 900 000 слов/с. Отличительной особенностью инструмента является наличие режима генерации слов по заданным морфологическим характеристикам [14].

Однако, при использовании библиотеки в различных прикладных информационных системах [6], а также для исследования и реализации алгоритмов обработки текстов, был выявлен ряд недостатков. Для

определения направления развития и повышения точно алгоритмов анализа в библиотеке JMorfSdk была проведена оценка точности их работы и сравнительный анализ с другими существующими инструментами морфологического анализа текстов на русском языке.

2. Сравнительный анализ работы инструментов морфологического анализа текстов на русском языке

На сегодняшний день самыми крупными корпусами текстов с морфологической разметкой являются Национальный корпус русского языка (НКРЯ) [3] и OpenCorpora [4].

В качестве корпуса для проведения анализа работы инструментов и последующего сравнения был выбран НКРЯ по причине большего количества размеченных слов. На момент анализа в корпусе находилось 95 056 предложений и 1 023 297 слов.

Сложность сравнения получения морфологических характеристик заключается в отсутствии для русского языка стандарта аннотации частей речи и остальных морфологических характеристик. Также словари некоторых инструментов содержат не весь набор характеристик, поэтому при сравнении для каждого анализатора были произведены сопоставления проставляемых тегов с анализируемым корпусом с минимальным набором из сравниваемых инструментов.

Был проведен сравнительный анализ работы следующих инструментов морфологического анализа:

- TreeTagger [7] – инструмент морфологического анализа, который использует словарь словоформ и словарь флексий. Последний предоставляет возможность проводить анализ слов, не входящих в его словарь. TreeTagger использует деревья решений для разрешения омонимии и требует размеченного корпуса для обучения;
- PullEnti [8] – библиотека, предоставляющая возможность полного анализа текста. Кроме того, в библиотеке реализована функция выделения именованных сущностей;
- pymorphy2 [9] – морфологический анализатор, за основу словаря которого взят OpenCorpora. Реализована вероятностная модель разрешения омонимии, а также средства анализа не словарных слов;
- RussianMorphology [10] – библиотека морфологического анализа, в которой взят за основу словарь из проекта AoT [11], базирующийся на грамматическом словаре Зализняка. Реализован анализ несловесных слов и разрешение омонимии;

– AoT [12] – библиотека морфологического анализа. Была использована новая версия на Java, в которой взят за основу проект AoT на C++.
– Natasha [13] – библиотеки для анализа текста. Позволяет произвести полный анализ текста в том числе и морфологический. Для проведения морфологического анализа используется библиотека pymorphy2.

Анализ инструментов производился по следующим параметрам:

1. Количество найденных слов – процент найденных слов от общего количества слов в корпусе.
2. Количество верно полученных начальной формы – процент верно определенных начальных форм слов от общего количества слов в корпусе.
3. 95-й процентиль времени получения начальной формы – ввиду того, что крайне сложно изолировать отдельно взятые процессы на ЭВМ, а также ввиду того, что на ЭВМ выполняется несколько процессов одновременно, то для того, чтобы минимизировать возможное влияние сторонних процессов на результат, используется 95 процентиль от всех измеряемых результатов. Для расчета используется метод ближайшего процентильного ранга.
4. Количество верно определенных морфологических характеристик без учета снятия омонимии – процент верно определенных морфологических характеристик без учета снятия омонимии от количества слов, найденных в словаре исследуемого инструмента. Верно определенными характеристики считается в случае, если полученные омоформы слов присутствуют и среди них есть хотя бы одна форма с верными характеристиками.
5. Количество верно определенных морфологических характеристик с учетом снятия омонимии – процент верно определенных морфологических характеристик с учетом снятия омонимии от количества слов, найденных в словаре анализатора. Верно определенными характеристики считается в случае, если полученная форма имеет верные характеристики, а в случае, если инструмент не может разрешить или не поддерживает снятие омонимии, то берется первая форма из омоформ.

Результаты оценки работы инструментов по количество найденных слов, верно полученных начальных форм и 95-ому процентилю времени получения начальной формы представлены в таблице 1.

В таблице 2 представлены результаты оценки точности получения морфологических характеристик с учетом и без снятия омонимии.

Таблица 1

Результаты оценки работы инструментов

Название инструмент	Количество найденных слов, %	Количество верно полученных начальных форм, %	95-й процентиль времени получения начальной формы, мс
JMorfSdk	95,6	82,3	0,001488
TreeTagger	99,0	92,3	0,269281
PullEnti	95,7	82,5	0,022700
pymorphy2	98,4	95,0	0,025552
RussianMorphology	94,8	90,4	0,001932
AoT	95,5	90,5	0,001644
Natasha	95,6	94,6	0,275177

Таблица 2

Результаты оценки точности получения морфологических характеристик с учетом и без снятия омонимии

Название инструмент	Количество верно определенных морфологических характеристик без учета снятия омонимии, %	Количество верно определенных морфологических характеристик с учетом снятия омонимии, %
JMorfSdk	90,8	70,2
TreeTagger	76,2	76,2
PullEnti	63,3	48,4
pymorphy2	91,7	76,6
RussianMorphology	67,6	63,0
AoT	96,8	72,8
Natasha	82,8	82,8

3. Проблемы качества результатов морфологического анализа текста

Анализ полученных результатов позволил выявить ряд проблем в работе инструментов.

В TreeTagger используется дерево принятия решения. По результатам имеет наименьший процент отсутствующих слов, однако, не производит анализ слов, написанных через дефис, возвращая саму словоформу в качестве начальной формы. Для некоторых глаголов, например, “доучила” некорректно происходит лемматизация, в результате чего возвращается сама словоформа в качестве начальной.

PullEnti в случае встречи омонимичной словоформы выводит все возможные варианты морфологических характеристик. В этой библиотеке не реализован механизм снятия морфологической омонимии. Кроме того, были обнаружены ошибки в процессе лемматизации – инструмент не имеет возможности для обработки слов, написанных через дефис, при этом приводя к начальной форме только первую часть словоформы. Также при поиске леммы у глаголов не учитывается возвратность, что приводит к неверной лемме у возвратных глаголов.

pymorphy2 показал наилучший результат по точности получения начальной формы, остальные параметры также являются близкими к наилучшим.

Библиотека Natasha показала наибольшую точность получения морфологических характеристик.

Проблемами анализаторов PullEnti, RussianMorphology, AoT и Natasha можно также считать небольшой по сравнению с остальными инструментами размер словаря, что снижает количество найденных слов.

В сравнительном анализе JMorfSdk с другими инструментами обработки текста были выявлены следующие недостатки:

- относительно высокий процент не найденных слов, что обусловлено отсутствием ёфицирования, т.е. инструмент использует словари с буквой «ё», в то, время как при письме зачастую буква «ё» заменяется на «е»;
- относительно невысокий процент точности получения начальных форм слов обусловлено тем, что при разработке инструмента были не учтены особенности используемого словаря, тем самым выдаются не инфинитивы глаголов, а их формы в первом лице или в мужском роде прошедшего времени;
- для каждого слова возвращаются все найденные формы, что требуется дополнительной обработки до перехода на синтаксический этап анализа.

4. Пути повышения точности алгоритмов морфологического анализа текстов в библиотеке JMorfSdk

На основе проведенного исследования были определены следующие пути повышения точности алгоритмов морфологического анализа текстов в библиотеке JMorfSdk:

Увеличение объема словаря.

Для более качественного морфологического анализа и передачи более точных данных на следующий, синтаксический, этап анализа текста важно увеличить количество найденных слов. Проект OpenCorpora, словарь которого использует в библиотеке JMorfSdk постоянно обновляется и обновление до текущей версии позволит частично решить эту проблему. Также дополнение словаря не найденными ранее словами может быть реализовано путем сбора отсутствующих слов при использовании библиотеки на большом объеме текстов из различных источников.

1. Улучшения уровня графематического анализа.

В текстах часто используются слова с опущенной буквой ё, поэтому необходимо либо добавление в словарь словоформ без ё с сохранением связей между словами, либо возможность ёфикации непосредственно во время проведения морфологического анализа. Выбор реализуемого способа зависит от приоритета по показателям работы библиотеки. Первый вариант приведет к увеличению объема используемой памяти из-за расширения словаря, второй - к увеличению длительности обработки из-за дополнительных преобразований. Добавление функции ёфикации позволит уменьшить количество не найденных слов с 4,39% до 4,18%.

JMorfSdk в отличие от некоторых других инструментов при проверке наличия в словаре не учитывает имена числительные в числовом представлении. Так как добавлять все числа в словарь нецелесообразно, то обозначение его как словарного и присвоение тега числительного позволят ещё уменьшить количество незнакомых слов в тексте, понизив количество не найденных слов суммарно до 2,23%.

2. Частичное разрешение омонимии.

Недостатком алгоритма анализа в текущей версии JMorfSdk является отсутствие функциональности снятия омонимии. На данный момент точность получения основных морфологических характеристик составляет 70,2%, при этом при рассмотрении всех наборов характеристик для омонимичных словоформ этот показатель может вырасти до 90,8%. Добавление даже бесконтекстного метода разрешения многозначности позволит значительно увеличить точность получения морфологических характеристик.

В результате обновления словаря, добавления новых словоформ и учет особенностей используемое словаря позволит повысить количество найденных слов до 98% и более.

Заключение

Стремительное развитие технологий приводит к тому, что с каждым годом не только появляются все больше различных инструментов обработки текста на естественном языке, но и развиваются существующие.

Анализ работы основных инструментов морфологического анализа текстов на русском языке показал, что не существует абсолютного лидера, которые имеет наилучшие результаты по всем параметрам. Однако, такой анализ позволил определить слабые места в JMorfSdk и определить пути повышения точности алгоритмов обработки текста для повышения точности морфологического анализа.

Так, например, подмена буквы «ё» на «е» при письме является очевидным, однако, не учитывалось в JMorfSdk. Также не учитывались особенности используемого словаря, что снижало качество получения начальных форм слов. Помимо этого, анализ показал, что необходимо решать задачу по снятию омонимии в момент морфологического анализа, а точнее вводить дополнительный этап пост-морфологического анализа, в рамках которого необходимо снимать неоднозначность слова как контекстными, так и бесконтекстными методами.

При этом, анализ подтверждает, что одна из задач, которая являлась приоритетной при разработке библиотеки JMorfSdk является выполненной, а именно высокая скорость анализа, в т.ч. получение начальной формы слова и его морфологических характеристик.

На основе проведенного исследования были определены пути повышения точности алгоритмов морфологического анализа текстов в библиотеке JMorfSdk, основными из которых являются расширение словаря, уточнения работы с данными на графематическом уровне и частичное разрешение омонимии.

Реализация алгоритмов в рамках определенных путей развития инструменты позволят существенно повысить точность работы алгоритмов морфологического анализа в библиотеке JMorfSdk и как следствие улучшить качество решения прикладных задач, связанных с обработкой текстов на русском языке.

Литература

1. Mohbey K. K., Tiwari S. Preprocessing and morphological analysis in text mining // International Journal of Electronics Communication and Computer Engineering. - 2011. - Vol. 2. - № 2. - P. 1-7.

2. Губин, М. В. Влияние морфологического анализа на качество информационного поиска / М. В. Губин, А. Б. Морозов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции : Труды Восьмой Всероссийской научной конференции (RCDL'2006), Суздаль, 17–19 октября 2006 года / Ярославский государственный университет им. П. Г. Демидова. – Суздаль: Ярославский государственный университет им. П.Г. Демидова, 2006. – С. 95-100.

3. Официальный сайт Национального корпуса русского языка. – Режим доступа – https://ruscorpora.ru. – (Дата обращения: 12.12.2021).

4. Официальный сайт OpenCorpora. – Режим доступа – http://opencorpora.org. – (Дата обращения: 12.12.2021).

5. Официальная страница JMorfSdk. – Режим доступа – https://github.com/jalexpr/jmorfsdk. – (Дата обращения: 12.12.2021).

6. Politsyna E., Politsyn S., Porechny A. Solving practical tasks of computer linguistics using the created text processing framework [Электронный ресурс]: статья. – Дата обращения: 15.12.2021 – Режим доступа: https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012129

7. Официальный сайт TreeTagger. – Режим доступа – https://cis.uni-muenchen.de/~schmid/tools/TreeTagger/. – (Дата обращения: 12.12.2021).

8. Официальный сайт PullEnti. – Режим доступа – https://pullenti.ru/. – (Дата обращения: 12.12.2021).

9. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts. - 2015. - P. 320-332.

10. Официальная страница RussianMorphology. – Режим доступа – https://github.com/AKuznetsov/russianmorphology. – (Дата обращения: 12.12.2021).

11. Официальный сайт AoT. – Режим доступа – http://aot.ru. – (Дата обращения: 12.12.2021).

12. Официальная страница AoT. – Режим доступа – https://github.com/demidko/aot. – (Дата обращения: 12.12.2021).

13. Официальная страница Natasha. – Режим доступа – https://github.com/natasha/natasha. – (Дата обращения: 12.12.2021).

14. Politsyna E. V. Development of the Cross-platform Library of Morphological Analysis of the Russian Language Text for Industrial Software / E. V. Politsyna, S. A. Politsyn, A. S. Porechny // CEE-SECR '18 Central and Eastern European Software Engineering Conference Russia Moscow. – ACM New York, NY, USA, 2018.