личаются род или число, то между ними нет связи. Также существуют правила третьего типа, которые указывают, какую пару слов следует предпочесть, если возможны несколько вариантов. Например, в предложении *Папа мыл окно* и «*папа*», и «*окно*» могут выступать в качестве субъекта, однако мы можем предпочесть предстоящее глаголу слово, а не последующее. Из описанного выше можно заметить, что такой подход очень ресурсоемок, так как для создания парсера необходима хорошая команда лингвистов, которой потребуется буквально описать весь русский язык. Поэтому в работе будем применять второй подход – машинное обучение с учителем.

Идея парсера, использующего машинное обучение, заключается в следующем. На вход классификатору подается много примеров с правильными ответами, на которых система должна обучиться самостоятельно. Чтобы обучить синтаксические классификаторы, в качестве данных для обучения используют специально размеченные корпуса, коллекции текстов, в которых размечена синтаксическая структура. Предложение, взятое для примера, может быть размечено так:

1	Папа	сущ.им.ед.муж.	2	субъект
2	мыл	глаг.ед.муж.прош.	0	–
3	машину	сущ.вин.ед.жен.	2	объект

В этом формате каждая строка описывает отдельное слово в виде записей. Для каждого слова нужно хранить следующие данные:

– номер слова в предложении (1);
– словоформа (папа);
– грамматические категории (сущ.им.ед.муж.);
– номер главного слова (2);
– тип связи (субъект).

Существует несколько открытых парсеров, которые можно обучить для работы с русским языком. Был выбран один из них, а именно MaltParser [9, 10]. Этот пакет включает в себя различные алгоритмы синтаксического анализа, в том числе алгоритмы Нивре и Ковингтона, а также реализации нескольких методов машинного обучения для предсказателей переходов. Благодаря своей эффективности и производительности MaltParser на данный момент является одним из наиболее широко используемых синтаксических анализаторов. Исследования по применению MaltParser проводились для многих языков, в том числе и для русского [11, 12]. Пакет может использоваться с различными алгоритмами обучения, с версии 1.3 по умолчанию поддерживаются два встроенных алгоритма обучения: LIBSVM [13] и LIBLINEAR [14]. По результатам обучения и тестирования второй алгоритм проявил себя заметно лучше, поэтому впоследствии все вычисления производились с ним.

Для обучения синтаксического классификатора использовался размеченный корпус СинТагРус, который входит в состав НКРЯ (данный корпус закрыт и предоставляется только для исследовательских целей). Корпус представляет собой набор текстов, размеченных в формате XML, с информацией о принадлежности каждого слова к части речи, соответствующих грамматических признаках, нормальной форме слова, синтаксических связях внутри предложения.

Для обучения синтаксичского анализатора необходимо перевести размеченные данные в формат malttab. Для этих целей был написан скрипт, который читает XML и выдает нужный формат, при этом нормализуя грамматические категории. Результат преобразования следующий:

Здесь	ADV	4	обст
уместнее	ADV.comp	4	обст
всего	S.gen.n.sg	2	сравнит
провести	VINF	0	ROOT
аналогию	S.acc.f.sg	4	1-компл
с	PR	5	1-компл
законами	S.ins.m.pl	6	предл
физики	S.f.gen.sg	7	квазиагент

Обучение MaltParser осуществляется на данных такого вида. В результате обучения получается файл *.mco, который содержит обученную модель и необходимые конфигурационные данные. Этот файл будет использоваться классификатором для синтаксического анализа неразмеченных данных. Общая схема анализатора и вспомогательных модулей представлена на рисунке 1.

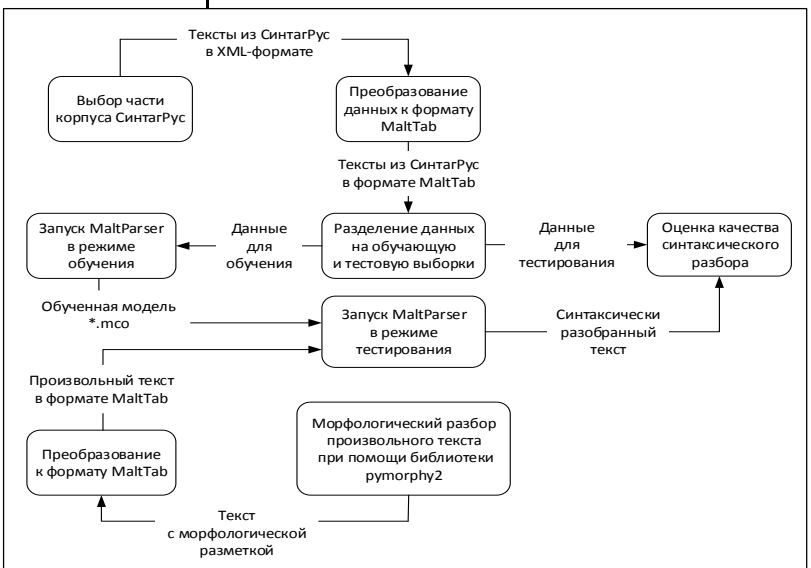

Рис. 1. Общая схема синтаксического анализатора

Fig. 1. General scheme of a parser

Одним из ключевых факторов, влияющих на оценку качества синтаксического анализатора, является результат морфологического парсера с использованием pymorphy2. Например, слово «ученый» в зависимости от контекста может быть как прилагательным, так и существительным, то есть является субстантивированным прилагательным (когда прилагательные переходят в разряд существительных). В подобных случаях неправильно определенная часть речи на уровне морфологического анализа влечет за собой ошибки на синтаксическом уровне. Также есть вероятность встретить в текстах слова-омонимы, частным случаем которых являются омоформы. Яркий пример – слово «печь», которое в зависимости от контекста выступает как существительное или глагол. Приняв ошибки такого рода, на основе предложенной модели получим результаты, представленные в таблице 1.

В таблице показаны результаты синтаксического анализатора MaltParser, обученного на 1/7 части корпуса СинТагРус. Для оценки качества результатов произведено несколько замеров, в таблице отражены усредненные значения. Для оценки полученной модели произведены замеры на морфологической разметке СинТагРус, которые дали неплохие результаты. Затем произведены замеры с использованием морфологической разметки из pymoprhy2. На основании результатов, отраженных в таблице 1, можно сделать вывод, что точность синтаксической разметки ухудшается.

Построение анафорического классификатора

В первую очередь происходит предобработка входного текста: он разбивается на предложения, после этого каждое предложение анализируется морфологическим и синтаксическим парсерами, результаты анализа для предложений сохраняются в файл и используются на следующих этапах. Затем анализатор просматривает текст слева направо, останавливаясь на каждом найденном местоимении. Для каждого местоимения текст просматривается в обратном направлении, в процессе чего составляется множество потенциальных антецедентов (кандидатов). В качестве кандидатов выступают морфологические интерпретации слов, то есть само слово и его морфологические характеристики. Как только такое множество составлено, анализатор продолжает просмотр текста в поисках следующего местоимения, затем анализатор выбирает из набора признаков единственного кандидата, которого считает верным антецедентом для данного местоимения. Таким образом, описываемый метод состоит из следующих этапов:

– поиск местоимения;

– составление множества потенциальных антецедентов;

– выбор наиболее подходящего кандидата.

Создание множества кандидатов. Основная цель данного этапа – создание множества морфологических интерпретаций, которые могут при некоторых условиях быть антецедентом для данного местоимения. Из полученного множества на следующих этапах выбирается единственный антецедент, он и будет признан верным. Такое множество должно удовлетворять критерию полноты, содержать в себе верный антецедент, чтобы у алгоритма выбора на последующих этапах была возможность его выбрать. В то же время представляется разумным ограничить пространство выбора и не включать в множество заведомо неподходящих кандидатов. Данный этап будем рассматривать как набор фильтров, которые будут применяться ко всем морфологическим интерпретациям, предшествующим в тексте местоимению. Каждый из фильтров должен либо отфильтровать кандидата, либо оставить его в рассматриваемом множестве.

Фильтр расстояния. Зачастую местоимение и антецедент находятся достаточно близко друг к другу. Хотя при тестировании и написании этого правила были зафиксированы случаи, когда антецедент и местоимение находились на расстоянии 8 предложений, тем не менее, большинство антецедентов укладываются в довольно маленькую окрестность местоимения. После проведения ряда экспериментов было решено ограничить рассматриваемое множество предложений на поиск претендентов до трех.

Морфологический фильтр. Морфологический фильтр, реализованный в данной работе, можно сформулировать с помощью набора правил:

– претендентами могут стать только существительные;

– кандидат и претендент должны быть согласованы в роде и числе.

Таблица 1

Результаты морфологического анализатора

Table 1

The results of a morphological parser

Выборка	Морфологическая разметка СинТагРус		Морфологическая разметка Pymorphy2	
	Точность (accuracy, %)	Время (сек.)	Точность (accuracy, %)	Время (сек.)
Выборка 2014 года Количество слов = 58 617 Количество предл. = 4 182	75	6,19	63	6,25
Выборка 2015 года Количество слов = 4 926 Количество предл. = 399	76	2,85	65	2,45

Синтаксический фильтр. Синтаксический фильтр, реализованный в данной работе, основан на обобщенных синтаксических ограничениях из статьи [15]. В работе показана их эффективность. Ограничения сформулированы как набор условий на взаимное расположение в синтаксическом дереве местоимения и кандидата. Условия описывают невозможные для местоимения и антецедента отношения, то есть, если хотя бы одно из условий выполняется, кандидат отвергается.

Для упрощения работы фильтра было решено объединить эти правила в одно. Чтобы претендент удовлетворял условиям фильтра, нужно, чтобы антецедент и местоимения принадлежали к разным Root-группам.

Пример, удовлетворяющий условиям фильтра:

Маша любит *мороженое*. Потому что **оно** вкусное.

1) Глагольная группа *любит*.
2) Root-группа *вкусное*.

Пример, не удовлетворяющий условиям фильтра:

Маше **она** нравится.

1) Глагольная группа *нравится*.
2) Root-группа *нравится*.

На этом этапе получен набор претендентов, удовлетворяющих условиям фильтра, это множество имеет небольшие размеры для конкурентоспособной работы классификатора.

Преобразование набора признаков к векторному виду. Затем на полученном множестве кандидатов ставится задача преобразования множества признаков в векторную форму. Набор признаков переводится в векторный формат по определенным правилам.

Расстояние в словах. Для каждого претендента высчитывается расстояние в словах до местоимения. В зависимости от этого расстояния вектор заполняется единицами. Были выделены три расстояния для фиксации их в векторе:

– от 10 слов включительно; вектор [1, 0, 0];
– от 10 до 30 слов включительно; вектор [0, 1, 0];
– от 30 слов; вектор [0, 0, 1].

Претенденту может соответствовать только один вектор с описанием в векторной форме.

Морфологические признаки. Из морфологических признаков, используемых в модели, для классификации учитывается только падеж:

– именительный [1, 0, 0, 0, 0, 0];
– родительный [0, 1, 0, 0, 0, 0];
– дательный [0, 0, 1, 0, 0, 0];
– винительный [0, 0, 0, 1, 0, 0];
– творительный [0, 0, 0, 0, 1, 0];
– предложный [0, 0, 0, 0, 0, 1].

Синтаксические признаки. В СинТагРус представлены четыре группы синтаксических отношений (СинтОтн) [16].

1. Актантные СинтОтн. Главной особенностью актантных СинтОтн является то, что они связывают предикатное слово [*X*] со словом [*Y*], заполняющим некоторую синтаксическую валентность этого предикатного слова. Всего 25 вариантов отношений.

2. Атрибутивные СинтОтн. Главной особенностью атрибутивных СинтОтн является то, что они связывают некоторое слово [*X*] со словом [*Y*], которое выражает при *X* значение невалентного атрибута – в самом широком смысле этого слова. Всего 31 вариант отношений.

3. Сочинительные СинтОтн. В принятой в данном корпусе системе синтаксических отношений сочинительные отношения принципиально не отличаются от подчинительных – и те, и другие связывают главный член конструкции с зависимым. Всего 5 вариантов отношений.

4. Служебные СинтОтн. Служебные СинтОтн связывают два элемента, синтаксически тесно связанные друг с другом. Часто члены таких конструкций – фактически части одного сложного слова. Всего 8 вариантов отношений.

Суммарно мы получаем набор из 69 видов отношений. Так как каждое слово может иметь только один тип отношений в предложении, вектор синтаксических признаков *X* будет иметь вид $X = [x_1, x_2, \ldots x_{69}]$, где $x_i \in \{0, 1\}$ и $\sum_{i=1}^{69} x_i = 1$.

Порядковый номер i определяется с помощью позиции в массиве отношений для каждого слова в зависимости от его синтаксических отношений внутри предложения.

Далее ставится задача комбинирования всех перечисленных признаков путем конкатенации векторов. Итоговый вектор *Z* представляется в виде $Z = [z_1, z_2, \ldots, z_{78}]$, где $z_i \in \{0, 1\}$ и $\sum_{i=1}^{78} z_i = 3$.

Таким образом, суммарное количество рассматриваемых признаков равно 78.

Описание работы классификатора. Для обучения анафорического классификатора использовался анафорически размеченный корпус текстов на русском языке, разработанный в Институте системного анализа ФИЦ ИУ РАН [6].

На основе размеченного корпуса текстов составляется обучающая выборка и строится бинарный классификатор, относящий с некоторой вероятностью кандидата к одному из двух классов: является антецедентом, не является антецедентом.

Обучающая выборка составляется следующим образом. Размеченные тексты просматриваются анализатором, для каждого местоимения составляется множество потенциальных антецедентов, для каждого кандидата из множества вычисляются описанные выше признаки. Далее вектор признаков каждого из кандидатов заносится в обучающую выборку с классом 1, если данный кандидат является антецедентом, и с классом 0 в противном случае.

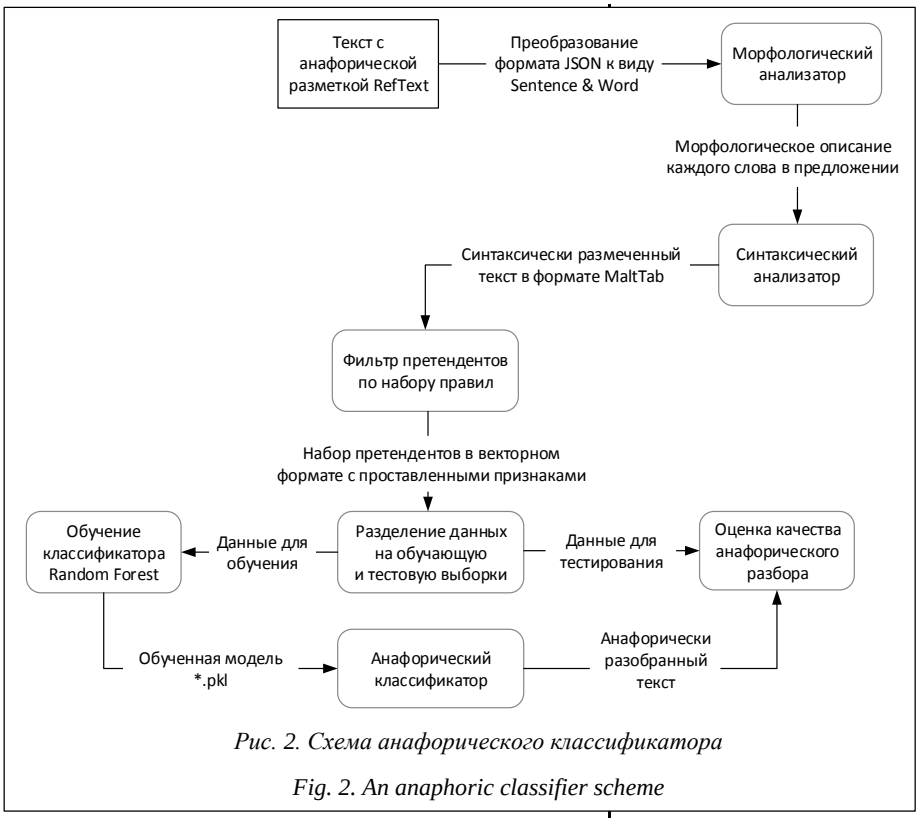

Рис. 2. Схема анафорического классификатора

Fig. 2. An anaphoric classifier scheme

В качестве алгоритма классификации был выбран алгоритм машинного обучения «решающий лес» (Random Forest) [17], основанный на деревьях принятия решений.

Сам по себе процесс построения каждого дерева можно описать следующим образом. Рассмотрим выборку X. Пусть x_j – j-й признак; t – порог значения признака; $x_j \le t$ – условие разбиения; $Q(X, j, t)$ – критерий, характеризующий ошибку разбиения. Требуется найти наилучшие параметры j и t, при которых ошибка разбиения будет минимальной.

Решающий лес (Random Forest) – ансамблевый алгоритм, представляющий собой множество решающих деревьев. Идея случайного леса состоит в том, чтобы выбрать наилучший признак j не из всех возможных признаков, а из случайного подмножества признаков некоторой заданной размерности. Выбор этого подмножества признаков осуществляется каждый раз при очередном разбиении вершины. Таким образом, каждое дерево $g_k(x)$ строится на своей обучающей выборке X'. В задаче классификации решение о выборе лучшего признака, по которому будет происходить разбиение, выбирается голосованием по большинству:

$$f(x) = sign \frac{1}{K} \sum_{k=1}^{K} g_k(x).$$

Данный подход к построению ансамбля моделей иногда называют бэггингом (bagging от Bootstrap Aggregating). Основная сложность его применения заключается в том, чтобы построенные модели действительно оказались независимыми.

Общая схема работы анафорического классификатора представлена на рисунке 2.

Полученные результаты и выводы

Суммарная точность считалась как процент правильно предсказанных антецедентов для местоимений, на которых производилось тестирование (табл. 2). Исследования проводились на следующих наборах признаков: Набор 1. Все признаки; Набор 2. Без падежей.

Проведенное тестирование показало, что описанный в работе метод может успешно применяться для разрешения местоименной анафоры, а также то, что качество анализатора улучшается, когда не учитывается морфологический фактор падежа.

Заключение

В данной работе предложен и реализован метод разрешения анафоры местоимений третьего лица в текстах на русском языке. Разработан алгоритм разрешения местоименной анафоры, включающий в себя следующие компоненты:

Таблица 2

Результаты анафорического классификатора

Table 2

The results of an anaphoric classifier

Количество деревьев	Аккуратность (accuracy, %)		F-мера (f1_score, %)		Точность (precision, %)		Полнота (recall, %)	
	Набор 1	Набор 2	Набор 1	Набор 2	Набор 1	Набор 2	Набор 1	Набор 2
1	57	64	53	58	45	52	64	66
5	64	64	56	59	55	53	57	67
10	62	64	55	56	50	52	60	66

– компонент создания множества кандидатов в антецеденты, основанный на применении набора фильтров: дистанционного, морфологического, синтаксического;

– классификатор выбора наиболее вероятного кандидата из множества предложенных на основе ряда признаков: дистанционного, морфологического, синтаксического.

Были проведены исследования по проверке эффективности данного метода, доказывающие применимость разработанного подхода.

Благодарность

Авторы работы выражают благодарность коллективу создателей синтаксически размеченного корпуса СинТагРус, разработанного в Институте проблем передачи информации РАН, и коллективу создателей анафорически размеченного корпуса текстов на русском языке, разработанного в Институте системного анализа РАН.

Литература

1. Шелманов А.О. Исследование методов автоматического анализа текстов и разработка интегрированной системы семантико-синтаксического анализа: дисс. … канд. тех. наук. М.: 2015. 210 с.

2. Barbu C., Mitkov R. Evaluation tool for rule-based anaphora resolution methods. Proc. 39th Annual Meeting on Association for Computational Linguistics, 2001, pp. 34–41.

3. Абрамов В.Е., Абрамова Н.Н., Некрасова Е.В., Росс Г.Н. Статистический анализ связности текстов по общественно-политической тематике // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. 13 Всерос. науч. конф. Воронеж, 2011. С. 127–133.

4. Толпегин П.В. Автоматическое разрешение кореференции местоимений третьего лица русскоязычных текстов: автореф. дисс. … канд. тех. наук. М., 2008. 241 с.

5. Protopopova E.V., Bodrova A.A., Volskaya S.A., Krylova I.V., Chuchunkov A.S., Alexeeva S.V., Bocharov V.V., Granovsky D.V. Anaphoric Annotation and Corpus-Based Anaphora Resolution: An Experiment, Computational Linguistics and Intellectual Technologies // Диалог-2014: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 13. С. 562–571 (англ.).

6. Kamenskaya M.A., Khramoin I.V., Smirnov I.V. Data-driven Methods for Anaphora Resolution of Russian // Диалог-2014: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 13. С. 241–250 (англ.).

7. Arregi O., et al. Determination of features for a machine learning approach to pronominal anaphora resolution in basque. Procesamiento del Language Natural, 2010, no. 45, pp. 291–294.

8. Korobov M. Morphological analyzer and generator for russian and ukrainian languages. Analysis of Images, Social Networks and Texts. 2015, pp. 320–332.

9. Nivre J., Hall J., Nilsson J. et al. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering. 2007, vol. 13, no. 2, pp. 95–135.

10. Nivre J., Hall J., Nilsson J. MaltParser: A data-driven parser-generator for dependency parsing. Proc. Intern. Conf. LREC, 2006, vol. 6, pp. 2216–2219.

11. Sharoff S., Nivre J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge // Диалог-2011: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 10. С. 591–604 (англ.).

12. Смирнов И.В., Шелманов А.О., Кузнецова Е.С., Храмоин И.В. Семантико-синтаксический анализ естественных языков. Ч. II. Метод семантико-синтаксического анализа текстов // Искусственный интеллект и принятие решений. М.: Изд-во ИСА РАН, 2014. № 1. С. 11–24.

13. Chang C.C., Lin C.J. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, vol 2, iss. 3, art. no. 27.

14. Fan R.E., Chang K.W., Hsieh C.J., Wang X.R. and Lin C.J. LIBLINEAR: A library for large linear classification Jour. of Machine Learning Research. 2008, vol. 9, pp. 1871–1874.

15. Lappin S., Leass H.J. An algorithm for pronominal anaphora resolution. Computational Linguistics. 1994, vol. 20, no. 4, pp. 535–561.

16. Национальный корпус русского языка. URL: http://www.ruscorpora.ru/instruction-syntax.html (дата обращения: 14.05.2017).

17. Breiman L. Random forests. Machine learning. 2001, vol. 45, no. 1, pp. 5–32.

Software & Systems
DOI: 10.15827/0236-235X.119.461-468

Received 15.05.17
2017, vol. 30, no. 3, pp. 461–468

ANAPHOR RESOLUTION SYSTEM DEVELOPMENT BASED ON MACHINE LEARNING METHODS

A.V. Sokolov [1], *Graduate Student, a23sokolov@gmail.com*
T.V. Batura [1, 2], *Ph.D. (Physics and Mathematics), Leading Researcher, Senior Researcher, tatiana.v.batura@gmail.com*

[1] *Novosibirsk State University, Pirogov St. 2, Novosibirsk, 630090, Russian Federation*
[2] *A.P. Ershov Institute of Informatics Systems (IIS), Siberian Branch of the Russian Federationn Academy of Sciences, Lavrentev Av. 6, Novosibirsk, 630090, Russian Federation*

Abstract. The paper proposes and implements a method for the anaphora resolution of third person pronouns in Russian texts.

The problem of finding the true pairs “anaphor-antecedent” is considered as a binary classification problem. Initially, the authors perform morphological and syntactic analysis of the text. The morphological analyzer used the pymorphy2 library. The parsing has been performed using MaltParser. The algorithm of anaphora resolution itself consists of three stages. First stage includes searching for all pronouns, then there is a compilation of many potential antecedents, and finally the most suitable candidate is selected. The component of creating a set of candidates for antecedents is based on using distance, morphological and syntactic filters. Classification uses the Random Forest algorithm. The anaphoric classifier takes into account 78 different features.

The authors performed a series of experiments in order to prove the effectiveness of the proposed method. They showed that that the quality of the analyzer improves if we do not take into account the morphological case. It can also be noted that the number of trees taken for calculation has a lesser effect on the final result when taking a feature set without cases.

The paper considers the main difficulties in developing the anaphora resolution systems. First, the search for anaphoric relations is in the semantic domain, and therefore it is difficult to formalize. Second, there are some features of the Russian language, such as developed morphology, morphological and syntactic ambiguities, which adversely affect the result.

Keywords: anaphora, antecedent, natural language processing, classification methods, machine learning, anaphoric classifier, analysis of text information.

Acknowledgements. *The authors express their gratitude to the team of creators of the syntactically marked corpus SinTagRus, which has been developed at the Institute for Information Transmission Problems of the Russian Academy of Sciences, and to the team of creators of the Anaphorically Marked Corpus of Texts in Russian developed at the Institute of System Analysis of the Russian Academy of Sciences.*

References

1. Shelmanov A.O. *Issledovanie metodov avtomaticheskogo analiza tekstov i razrabotka integrirovannoy sistemy semantiko-sintaksicheskogo analiza* [A Research on Automatic Text Analysis Methods and a Development of an Integrated System of Semantic-Syntactic Analysis]. PhD Thesis. Moscow, 2015, 210 p.
2. Barbu C., Mitkov R. Evaluation tool for rule-based anaphora resolution methods. *Proc. 39th Annual Meeting on Association for Computational Linguistics*. 2001, pp. 34–41.
3. Abramov V.E., Abramova N.N., Nekrasova E.V., Ross G.N. Statistical analysis of the connectivity of texts on social and political topics. *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii: tr. 13 Vseros. nauch. konf.* [Proc. 13th All-Russian Science Conf. "Digital Libraries: Advanced Methods and Technologies, Digital Collections" (RCDL'2011)]. 2011, pp. 127–133 (in Russ.).
4. Tolpegin P.V. *Avtomaticheskoe razreshenie koreferentsii mestoimeny tretyego litsa russkoyazychnykh tekstov* [Automatic resolution of the co-reference of third person pronouns in Russian-language texts]. PhD Thesis. Moscow, 2008, 214 p. (in Russ.).
5. Protopopova E.V., Bodrova A.A., Volskaya S.A., Krylova I.V., Chuchunkov A.S., Alexeeva S.V., Bocharov V.V., Granovsky D.V. Anaphoric annotation and corpus-based anaphora resolution: an experiment, computational linguistics and intellectual technologies. *Dialog-2014: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistecs "Dialogue-2014"]. 2014, iss. 13 (20), pp. 562–571 (in Russ.).
6. Kamenskaya M.A., Khramoin I.V., Smirnov I.V. Data-driven Methods for Anaphora Resolution of Russian. *Dialog-2014: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistecs "Dialogue-2014"]. 2014, iss. 13 (20), pp. 241–250 (in Russ.).
7. Arregi O., Ceberio K., Díaz de Illarraza A., Goenaga I., Sierra B., Zelaia A. Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque. *Procesamiento del language natural* [Natural Language Processing]. 2010, no. 45, pp. 291–294.
8. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages. *Analysis of Images, Social Networks and Texts*. 2015, pp. 320–332.
9. Nivre J., Hall J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marinov S., Marsi E. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*. 2007, vol. 13, no. 2, pp. 95–135.
10. Nivre J., Hall J., Nilsson J. MaltParser: A data-driven parser-generator for dependency parsing. *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*. 2006, vol. 6, pp. 2216–2219.
11. Sharoff S., Nivre J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. *Dialog-2011: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistecs "Dialogue-2011"]. 2011, no. 10, pp. 591–604.
12. Smirnov I.V., Shelmanov A.O., Kuznetsova E.S., Khramoin I.V. Semantic-syntactic analysis of natural languages. Part II. Method of semantic-syntactic analysis of texts. *Iskusstvenny intellekt i prinyatie resheny* [Artificial intelligence and decision-making]. Moscow, 2014, no. 1, pp. 11–24 (in Russ.)
13. Chang C.C., Lin C.J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. on Intelligent Systems and Technology*. 2011, vol. 2, iss. 3, art. no. 272001.
14. Fan R.E., Chang K.W., Hsieh C.J., Wang X.R., Lin C.J. *LIBLINEAR: A library for large linear classification Journal of Machine Learning Research*. 2008, vol. 9, pp. 1871–1874.
15. Lappin S., Leass H.J. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*. 1994, vol. 20, no. 4, pp. 535–561.
16. *Natsionalny korpus russkogo yazyka* [Russian National Corpus]. Available at: http://www.ruscorpora.ru/instruction-syntax.html (accessed May 14, 2017).
17. Breiman L. Random forests. *Machine learning*. 2001, vol. 45, no. 1, pp. 5–32.