

УДК 004.01:006.72 (470.22)

## ОБЗОР МЕТОДОВ (УТОЧНИТЬ - КАКИХ?) ПО РЕШЕНИЮ ЗАДАЧИ РАЗРЕШЕНИЯ ЛЕКСИЧЕСКОЙ МНОГОЗНАЧНОСТИ

С. С. Ткач, Е. А. Ярышкина, А. Н. Кирилов, И. А. Сихонина, Н. И. Коржицкий, А. М. Спирикова

*Институт прикладных математических исследований Карельского научного центра РАН*

Данный файл является примером статьи для научного издания Труды Карельского научного центра РАН, серия «Математическое моделирование и информационные технологии». В нем содержатся основные используемые переменные и окружения. При подготовке статьи рекомендуется воспользоваться этим примером в качестве шаблона. Данный абзац оформлен в стиле аннотации статьи.

Ключевые слова: труды, шаблон, подготовка статьи.

S. S. Tkach, E. A. Yaryshkina, A. N. Kirilov, I. A. Sikhonina,  
N. I. Korzhitsky, A. M. Spirikova. WORD-SENSE  
DISAMBIGUATION METHODS (SPECIFIC?) REVIEW

This file is an auxiliary example of an article prepared for Transactions of Karelian Research Centre of RAS. It contains most useful environments and variables. While preparing your article, it's recommended to use this text as a template. This paragraph is formatted as an Abstract of the article.

Key words: transactions, template, article.

---

### ВВЕДЕНИЕ

Статья, представляемая в научное издание, должна быть оформлена в соответствии с «Правилами для авторов», размещенными на сайте <http://transactions.krc.karelia.ru>. Данный документ претендует на роль технической документации в помощь авторам. Достаточно подробную информацию по набору в системе  $\LaTeX$  можно найти, напр., в работе [2].

### СТРУКТУРА ФАЙЛА В ФОРМАТЕ $\LaTeX$ 2 $\epsilon$

```
\documentclass{article}
\usepackage{krctran}
...
\begin{document}
```

```
\procname{...}
\udk{...}
\rustitle{...}
\engtitle{...}
\rusauthor{...}
\engauthor{...}
\organization{...}
\rusabstract{...}
\engabstract{...}
\ruskeywords{...}
\engkeywords{...}
```

```
\maketitle

\begin{articletext}
```

Рис. 1. Задержки в модели 10-узловой системы, случай тяжелых хвостов

```
\section{...}
...
\begin{thebibliography}
...
\end{thebibliography}
\end{articletext}

\section{СВЕДЕНИЯ ОБ АВТОРЕ:}
\begin{aboutauthors}
...
\end{aboutauthors}
\end{document}
```

Преамбула статьи должна содержать две обязательные команды:

```
\documentclass{article}
\usepackage{krctran}
```

Далее формируется заголовок статьи: выходные данные, код УДК, название статьи,

авторы с указанием мест работы, аннотация, ключевые слова. Например, заголовок этого файла сформирован следующими командами:

```
\procname{Труды...\ \No...}
\udk{УДК...}
\rustitle{Руководство...}
\engtitle{Usage...}
\rusauthor{А.~С.~Румянцев}
\engauthor{A.~S.~Rumyantsev}
\organization{Институт...}
\rusabstract{Данный файл...}
\engabstract{This file...}
\ruskeywords{труды,...}
\engkeywords{transactions,...}
\maketitle
```

**Замечание 1.** Для ручной разбивки на строки названия статьи воспользуйтесь командой `\newline`.

## WSD-МЕТОДЫ С УЧИТЕЛЕМ

### «СРАВНИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ В WSD: РОЛЬ ПРЕДПОЧТЕНИЙ В МАШИННОМ ОБУЧЕНИИ»

*Н. И. Коржицкий*

В работе Рэймонда Муни [20] представлено одно из первых сравнений разных по природе методов WSD на одних и тех же данных. В статье [20] проведена серия экспериментов, в которых сравнивалась способность различных обучающихся алгоритмов определять значение слова в зависимости от контекста. Эксперимент состоял в определении одного из шести значений слова *line* в различных контекстах. Данные для проведения экспериментов взяты из работы [15].

В машинном обучении под термином *bias* (пристрастие, тенденция, предпочтение) понимается любое основание для выбора одного обобщения другому, вместо строгого соответствия примерам [20]. В деревьях принятия решений предпочтение (*bias*) отдаётся простым деревьям решений, в нейронных сетях – линейным пороговым функциям, а в байесовском классификаторе – функциям, учитывающим условную независимость свойств. Чем лучше “предпочтение” обучающегося алгоритма соответствует характеристикам конкретной задачи, тем лучше будет результат. Большинство обучающихся алгоритмов обладают

“предпочтением” наподобие Бритвы Оккама, в таких алгоритмах выбираются гипотезы, которые могут быть представлены меньшим количеством информации на каком-нибудь языке представлений. Однако компактность, с которой (деревья решений, дизъюнктивная нормальная форма, сети с линейным пороговым значением) представляют конкретные функции – может существенно различаться. Поэтому различные “предпочтительные” оценки могут работать лучше или хуже в конкретных задачах. Одной из основных целей в машинном обучении является поиск “предпочтений” с целью решения прикладных практических задач.

Выбор правильного “предпочтения” и обучающегося алгоритма является сложной задачей. Простым подходом является автоматизация выбора метода при помощи внутренней перекрестной валидации. Другой подход *meta-learning* заключается в том, чтобы обучиться набору правил (или другому классификатору), предсказывающему, когда обучающийся алгоритм будет срабатывать наилучшим образом на примере с набором свойств присущих проблеме.

Описанный в [20] эксперимент заключается в определении значения слова *line* (англ. линия) среди 6 возможных вариантов (строка, ряд, дивизия, телефон, веревка, продуктовая линия).

Для получения обучающей выборки брались предложения со словом *line*, и им в со-

ответствие ставилось одно из 6 значений. Распределение значений неравномерно: включе-

ние в список источников журнала The Wall Street Journal привело к тому, что одно из значений встречалось в 5 раз чаще остальных [15].

Таблица 1. Шесть значений слова *line* из Английского Викисловаря и Русского Викисловаря

ключевое слово	перевод	толкование на английском (Английский Викисловарь)	толкование на русском (Русский Викисловарь)
text	строка	A small amount of text	ряд слов, букв или иных знаков, написанных или напечатанных в одну линию
formation	ряд	A more-or-less straight sequence of people, objects, etc., either arranged as a queue or column and often waiting to be processed or dealt with, or arranged abreast of one another in a row (and contrasted with a column), as in a military formation	несколько объектов, расположенных в линию или следующих один за другим
division	дивизия	A formation, usually made up of two or three brigades	тактическое воинское соединение
phone	телефон	The wire connecting one telegraphic station with another, a telephone or internet cable between two points: a telephone or network connection	то же, что телефонный номер
cord	веревка	A rope, cord, string, or thread, of any thickness	гибкое и длинное изделие, — чаще всего сплетённое или свитое из льняных (или пеньковых, полимерных и т. п.) волокон или прядей
product	продуктовая линия	The products or services sold by a business, or by extension, the business itself	совокупность однородной продукции единого назначения

В работе [11] было установлено, что наиболее эффективными при решении задачи WSD являются алгоритмы на основе дерева решений (decision tree). Данный класс методов обходился по точности и скорости работы класс нейронных сетей. Другие исследования [21] показали, что класс методов индуктивного логического программирования (inductive logic programming) справляется с задачей разрешения лексической многозначности слова лучше алгоритмов на основе дерева решений.

В серии экспериментов в [20] сравнивались следующие методы: байесовский классификатор, перцептрон, C4.5, метод k-ближайших соседей и модификации алгоритма FOIL: PFOIL-DLIST, PROIL-DNF, PFOIL-CNF. Все алгоритмы были реализованы на языке

Common Lisp, за исключением C4.5, который был написан на языке C.

После проведения сравнительных экспериментов, заключавшихся в обучении и определении значения слова *line*, было выяснено, что байесовский классификатор и перцептрон работают точнее других рассмотренных методов.

Эксперименты проводились с разными размерами обучающей выборки для того, чтобы выяснить, какого рода зависимость имеет место между точностью определения значения и размером выборки. На рисунке 2 отображена зависимость точности работы алгоритмов от размера выборки. При увеличении размера обучающей выборки сначала происходит резкий рост точности, последующий прирост точности становится незначительным.

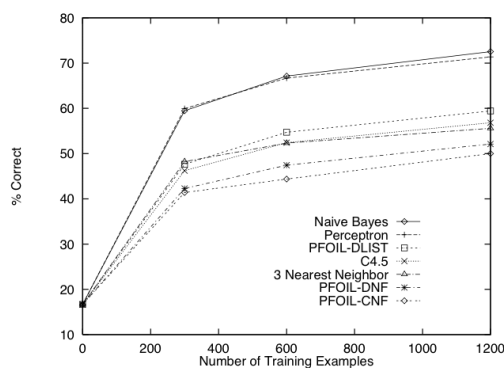


Рис. 2. Рост точности решения WSD задачи для разных алгоритмов (определение значения слова *line*) при увеличении размера обучающей выборки [20]. Алгоритмы: PFOIL-DLIST, PROIL-DNF, PFOIL-CNF, C4.5, Naive Bayes – наивный байесовский классификатор; Perceptron – перцептрон; 3 Nearest Neighbor – метод 3-х ближайших соседей

Эксперименты учитывали не только точность определения значения, но и требовательность алгоритма к ресурсам в процессе обучения и работы. На рисунке 3 можно увидеть зависимость времени обучения от размера выборки. Самыми быстрообучаемыми оказались байесовский классификатор и перцептрон, а самыми медленными – нормальные формы (рис. 3).

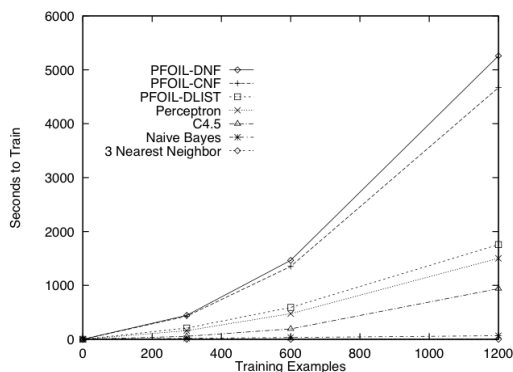


Рис. 3. Зависимость времени затраченного на обучение алгоритмов от размера обучающей выборки [20]. Алгоритмы: PFOIL-DLIST, PROIL-DNF, PFOIL-CNF, C4.5, Naive Bayes – наивный байесовский классификатор; Perceptron – перцептрон; 3 Nearest Neighbor – метод 3-х ближайших соседей

На рисунке 4 представлена зависимость времени работы алгоритмов от размера обучающей выборки. Время работы алгоритмов дает другую картину: байесовский классификатор и перцептрон работают долго при максимальном размере обучающей выборки, в то

время как остальные методы решают WSD задачу за постоянное время (рис. 4).

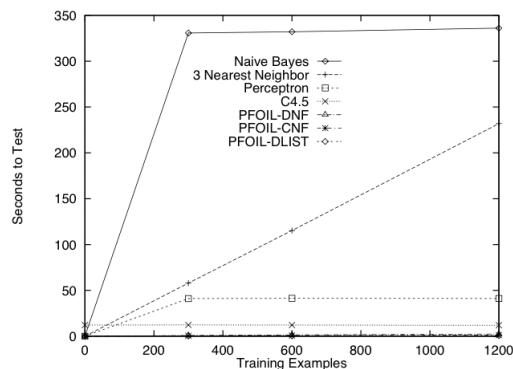


Рис. 4. Зависимость времени работы алгоритмов от размера обучающей выборки при определении значения слова *line* [20]. Алгоритмы: PFOIL-DLIST, PROIL-DNF, PFOIL-CNF, C4.5, Naive Bayes – наивный байесовский классификатор; Perceptron – перцептрон; 3 Nearest Neighbor – метод 3-х ближайших соседей

## «РАЗРЕШЕНИЕ ЛЕКСИЧЕСКОЙ МНОГООЗНАЧНОСТИ МЕТОДОМ АНСАМБЛЯ БАЙЕСОВСКИХ КЛАССИФИКАТОРОВ»

А. Л. Чухарев, Т. В. Каушинис

В статье [23] рассматривается подход к разрешению лексической многозначности слов (WSD), подразумевающий создание ансамбля Наивных Байесовских классификаторов, каждый из которых основан на оценке вероятности вхождения определенных слов в контекст целевого слова, значение которого определяется.

При разрешении лексической многозначности, представленной как задача обучения с учителем, применяют статистические методы и методы машинного обучения к размеченному корпусу. В таких методах словам корпуса, для которых указано значение, соответствует набор лингвистических свойств. Алгоритм обучения строит модель классификатора значений по этим лингвистическим особенностям.

Автор статьи [23] предлагает подход, основанный на объединении ряда простых классификаторов в ансамбль, который разрешает многозначность с помощью голосования простым большинством голосов. Педерсен утверждает [23], что, во-первых, рассмотрение более сложного набора языковых особенностей или более сложных алгоритмов обычно не улучшает точность разрешения по сравнению с про-

стыми языковыми особенностями, используемых в алгоритме обучения с учителем. Вторых, совместная встречаемость слов и словосочетаний имеет большее влияние на точность разрешения, оперирование такой лингвистической информацией как: часть речи или наличие отношений действие-объект.

В рассматриваемой статье [23] в ансамбль объединяются Наивные Байесовские классификаторы. При таком подходе предполагается, что все переменные, участвующие в представлении проблемы, – условно независимы, при фиксированном значении переменной классификации. В проблеме разрешения лексической многозначности существует понятие контекста, в котором встречается многозначное слово. Он представляется в виде функции переменных  $(F_1, F_2, \dots, F_n)$ , а значение многозначного слова представлено в виде классификационной переменной ( $S$ ). Все переменные бинарные. Переменная, соответствующая слову из контекста, принимает значение ИСТИНА, если это слово находится на расстоянии определенного количества слов слева или справа от целевого слова. Совместная вероятность наблюдения определённой комбинации переменных контекста с конкретным значением слова выражается следующим образом:  $p(F_1, F_2, \dots, F_n S) = (p(S) \prod_{i=1}^n p(F_i \vee S))$ , где  $p(S)$  и  $p(F_i|S)$  – параметры данной модели. Для оценки параметров достаточно знать частоты событий, описываемых взаимозависимыми переменными  $(F_i, S)$ . Эти значения соответствуют числу предложений, где слово, представляемое  $F_i$ , встречается в некотором контексте многозначного слова, упомянутого в значении  $S$ . Если возникают нулевые значения параметров, то они сглаживаются путём присвоения им по умолчанию очень маленького значения. После оценки всех параметров модель считается обученной и может быть использована в качестве классификатора.

Контекст в [23] представлен в виде bag-of-words (модель «мешка слов»). В этой модели выполняется следующая предобработка текста: удаляются знаки препинания, все слова переводятся в нижний регистр, все слова приводятся к их начальной форме (лемматизация). В [23] контексты делятся на два окна: левое и правое. В первое попадают слова, встречающиеся слева от неоднозначного слова, и, соответственно, во второе – встречающиеся справа.

Окна контекстов могут принимать 9 различных размеров: 0, 1, 2, 3, 4, 5, 10, 25 и 50 слов. Первым шагом в ансамблевом подходе является обучение отдельных Наивных Бай-

есовских классификаторов для каждого из 81 возможных сочетаний левого и правого размеров окон. В статье [23] Наивный Байесовский классификатор  $(l, r)$  включает в себя  $l$  слов слева от неоднозначного слова и  $r$  слов справа. Исключением является классификатор  $(0,0)$ , который не включает в себя слов ни слева, ни справа. В случае нулевого контекста классификатору присваивается **априорная вероятность** многозначного слова (равная вероятности встретить наиболее употребимое значение).

Следующий шаг в [23] при построении ансамбля – это выбор классификаторов, которые станут членами ансамбля. 81 классификатор группируется в три общие категории, по размеру окна контекста. Используются три таких диапазона: узкий (окна шириной в 0, 1 и 2 слова), средний (3, 4, 5 слов), широкий (10, 25, 50 слов). Всего есть 9 возможных комбинаций, поскольку левое и правое окна отделены друг от друга. Например, Наивный Байес (1,3) относится к диапазону категории (узкий, средний) поскольку он основан на окне из одного слова слева и окне из трёх слов справа. Наиболее точный классификатор в каждой из 9 категорий диапазонов выбирается для включения в ансамбль. Затем каждый из 9 членов классификаторов голосует за наиболее вероятное значение слова с учетом контекста. После этого ансамбль разрешает многозначность путем присвоения целевому слову значения, получившего наибольшее число голосов.

**Экспериментальные данные.** Для экспериментов были выбраны английские слова *line* и *interest*. Источником статистических данных по этим словам послужили работы [15], [8]. В статье приводится информация о частоте использования шести значений для каждого из этих слов (Табл. 2, Табл. 3).

Таблица 2. Число употреблений слова *line* (столбец *count*) для шести наиболее часто встречаемых значений (значения из тезауруса WordNet, столбец *sense*) по данным корпусов *ACL/DCI Wall Street Journal* и *American Printing House for the Blind*

sense	count
product	2218
written or spoken text	405
telephone connection	429
formation of people or things; queue	349
an artificial division; boundary	376
a thin, flexible object; cord	371
total	4148

Таблица 3. Число употреблений слова *interest* (столбец *count*) для шести наиболее часто встречаемых значений (значения из словаря Longman Dictionary of Contemporary English, столбец *sense*). Этот набор данных был получен в 1994 году Брюсом и Виебе [8] путём указания значений для всех вхождений слова *interest* в корпус ACL/DCI Wall Street Journal

sense	count
money paid for the use of money	1252
a share in a company or business	500
readiness to give attention	361
advantage, advancement or favor	178
activity that one gives attention to	66
causing attention to be given to	11
total	2368

**Результаты экспериментов.** Итогом проделанной работы стали обучение и проверка 81 Наивного Байесовского классификатора на многозначных словах *line* и *interest*. Точность разрешения лексической многозначности составила 89% для слова *interest* и 88% для слова *line*. В [23] было получено, что ансамбль классификаторов с голосованием простым большинством даёт более высокую точность, чем взвешенное голосование. Например, для слова *interest* при голосовании простым большинством точность составила 89%, а взвешенное голосование дало только 83%.

#### «WSD НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ, ПОСТРОЕННЫХ ПО ДАННЫМ МАШИНО-ЧИТАЕМЫХ СЛОВАРЕЙ»

А. Н. Кирилов

Настоящий текст является рефератом статьи [28], в которой описан метод автоматического построения очень больших нейронных сетей (VLNN) с помощью текстов, извлекаемых из машинно-читаемых словарей (MRD), и рассмотрено использование этих сетей в задачах разрешения лексической неоднозначности (WSD).

В дальнейшем будем называть слова, смысл которых требуется установить целевыми словами.

Широко известен метод Леска [16] использования информации из MRD для задачи WSD. Суть этого метода состоит в вычислении так называемой «степени пересечения», т.е. количества общих слов в словарных определениях слов из контекста («окна») условного размера, содержащего целевое слово. Основной недостаток метода Леска – зависимость от словарной статьи, т.е. от слов, входящих

в нее. Стратегия преодоления этого недостатка – использование словарных статей, определяющих слова, входящие в другие словарные статьи, начиная со словарных статей, соответствующих словам из контекста. Таким образом, образуются достаточно длинные пути из слов, входящих в словарные статьи. Эта идея лежит в основе топологии (строения) VLNN.

Использование нейронных сетей для WSD было предложено в работах [10, 32]. В рассматриваемой статье для построения VLNN использован словарь Collins English Dictionary.

**Топология сети.** Целевое слово представлено узлом, соединенным активирующими связями со смысловыми узлами, представляющими все возможные смыслы слова, имеющиеся в словарных статьях. Каждый смысловой узел, в свою очередь, соединен активирующими связями с узлами, представляющими слова в словарной статье, соответствующей определению данного смысла. Процесс соединения повторяется многократно, создавая большую сеть взаимосвязанных узлов. В идеале сеть может содержать весь словарь. Авторы, по практическим соображениям, ограничиваются несколькими тысячами узлов и 10 – 20 тысячами соединений. Слова представлены своими леммами (каноническими формами). Узлы, представляющие различные смыслы данного слова, соединены запрещающими (подавляющими) связями.

**Алгоритм функционирования сети.** При запуске сети первыми активируются узлы входного слова, которое кодируется согласно принятому правилу. Затем каждый входной узел посылает активирующий сигнал своим смысловым узлам, с которыми он соединен. В результате сигналы распространяются по всей сети в течение определенного числа циклов. В каждом цикле узлы слова и его смыслов получают обратные сигналы от узлов, соединенных с ними. Узлы конкурирующих смыслов посылают взаимно подавляющие сигналы. Взаимодействие сигналов обратной связи и подавления, в соответствии со стратегией «победитель получает все», позволяет увеличить активацию узлов-слов и соответствующих им правильных узлов-смыслов, одновременно уменьшая активацию узлов соответствующих неправильным смыслам. После нескольких десятков циклов сеть стабилизируется в состоянии, в котором активированы только узлы-смыслы с наиболее активированными связями с узлами-словами. В статье не указан алгоритм настройки, т.е. обучения сети. Видимо, используется метод встречного распространения (back propagation).

### «РАЗЛИЧЕНИЕ ЗНАЧЕНИЙ СЛОВ НА ОСНОВЕ ВЕКТОРОВ СВОЙСТВ, РАСШИРЕННЫХ СЛОВАРНЫМИ ТОЛКОВАНИЯМИ»

А. М. Спирикова

Амрута Пурандаре и Тед Педерсен в 2004 году разработали "Алгоритм различения значений на основе контекстных векторов" (Context vector sense discrimination) [25]. В этом алгоритме (1) берется набор примеров употреблений исследуемого слова, (2) выполняется кластеризация этих примеров так, чтобы близкие по значению или связанные каким-либо образом слова объединились в одну группу [25].

*Различение значений слов* (word sense discrimination) – это задача группировки нескольких употреблений данного слова в кластеры, где каждому кластеру соответствует определенное значение целевого слова. Подходы к решению этой проблемы основываются на дистрибутивной гипотезе, которая говорит о том, что: лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения. Следует различать понятия *различение значений слов* и *разрешение лексической многозначности*. При *различении значений слов* нет никаких предопределённых значений слова, присоединенных к кластерам; здесь, скорее, слова, употребляющиеся в схожих контекстах, группируются в кластеры (значения).

При решении задачи *различения значений* используются контекстные вектора: если целевое слово встречается в тестовых данных, то контекст этого слова представляется в виде вектора контекста. *Вектор контекста* – это средний вектор по векторам свойств каждого из слов контекста. *Вектор свойств* содержит информацию о совместной встречаемости данного слова с другими словами, этот вектор строится по данным корпуса текстов на этапе обучения.

Метод различения значений Пурандаре и Педерсена [25] предназначен для работы при недостаточном объёме текстовых данных, при этом вектор свойств расширяется данными, извлечёнными из толкований словарей. Этот

метод группирует в кластеры близкие по значению употребления целевого слова.

**Построение матрицы встречаемости слов.** Первоначально строится матрица совместной встречаемости слов по данным обучающего корпуса (здесь тексты Wall Street Journal и Британского национального корпуса).

Вектор свойств (строка матрицы) содержит информацию о совместной встречаемости данного слова с другими. Было решено в [25], что слова «встречаются», если они находятся в тексте на расстоянии не более пяти словопозиций (то есть между ними находится не более трёх слов).

**Обработка матрицы.** После создания матрицы выполняется разделение тестовых данных, то есть группировка примеров употреблений (фраз) с целевым словом. Каждому слову в примере употребления в тестовых данных соответствует вектор свойств из матрицы встречаемости. Средний вектор свойств по всем словам соответствует вектору контекста. Таким образом, набор тестовых данных, включающих употребление исследуемого слова, преобразуется в набор контекстных векторов, каждый из которых соответствует одному из употреблений целевого слова.

Различение значений происходит путем кластеризации контекстных векторов с помощью разделяющего (partitional) или иерархического «сверху вниз» (agglomerative) алгоритма кластеризации [14], [13], [34]. Получающиеся кластеры составлены из употреблений близких по значению фраз, и каждый кластер соответствует отдельному значению целевого слова.

**Векторы свойств, расширенные текстами толкований из словаря.** Векторы свойств, полученные по небольшому корпусу текстов, имеют очень малую размерность (несколько сотен), что не позволяет полностью описать закономерности совместной встречаемости слов. Для решения этой проблемы векторы свойств слов расширяются содержательными словами (content words), извлечёнными из словарных толкований разных значений данного слова. В таблице 4 представлены примеры толкований и содержательные слова для восьми значений слова «история» из Русского Викисловаря.

Таблица 4. Словарные толкования (и содержательные слова) по данным статьи «история» из Русского Викисловаря. Серым цветом и курсивом выделены те слова, которые уже были в векторе слов, чёрным – новые слова из толкований, которыми будет расширен вектор

№	Текст значения	Содержательные слова
1	закономерное, последовательное развитие, изменение действительности	<i>развитие</i> , изменение
2	наука, изучающая факты, тенденции и закономерности развития человеческого общества	<i>наука</i> , факт, тенденция, закономерность
3	наука, изучающая ход развития, последовательные изменения какой-либо области природы или культуры	<i>наука</i> , <i>развитие</i> , изменение
4	последовательный ход развития, изменения чего-либо, совокупность фактов о развитии какого-либо явления	<i>развитие</i> , изменение, факт
5	отдалённое время с его событиями, происшествиями; прошлое	время, событие, происшествие
6	эпическое повествование, рассказ	повествование, <i>рассказ</i>
7	смешная или неожиданная ситуация, происшествие, случай	ситуация, случай, происшествие
8	скандал, неприятность	скандал, неприятность

Предположим, например, что вектор свойств (столбец в матрице встречаемости) для слова *история* имеет непустые значения в строках, соответствующих словам: *книга*, *мир*, *наука*, *образование*, *развитие*, *рассказ*.

В русском Викисловаре различные значения слова *история* (таблица 4) включают содержательные слова: *время*, *закономерность*, *изменение*, *наука*, *неприятность*, *повествование*, *происшествие*, *развитие*, *рассказ*, *ситуация*, *скандал*, *случай*, *событие*, *тенденция*, *факт*. Таким образом, вектор свойств, соответствующий слову «история», будет расширен новыми (отсутствующими ранее) словами из словаря: *время*, *закономерность*, *изменение*, *неприятность*, *повествование*, *происшествие*, *ситуация*, *скандал*, *случай*, *событие*, *тенденция*, *факт*.

В итоге, вектор свойств будет включать слова: *время*, *закономерность*, *изменение*, *книга*, *мир*, *наука*, *неприятность*, *образование*, *повествование*, *происшествие*, *развитие*, *рассказ*, *ситуация*, *скандал*, *случай*, *тенденция*, *факт*.

Для оценки результатов тестовым примерам употребления присваивали вручную теги значений. Кластеру присваивалось то значение, примеров употребления которого в нём было больше всего.

Авторами было проведено 75 экспериментов с использованием 72 слов из корпуса SENSEVAL-2 и со словами “line”, “hard” и “serve”.

В тестовых данных SENSEVAL-2 примеры употреблений включали 2-3 предложения. Для каждого слова было дано от 50 до 200

примеров употреблений в тестовых и тренировочных данных. Для этих слов известно много (порядка 8-12) значений. Малое число примеров при большем числе значений привело к тому, что для некоторых значений оказалось мало примеров употреблений. 43 из 72 слов SENSEVAL-2 показали улучшение F-меры и полноты (recall) при расширении вектора свойств текстами толкований словаря. Однако для 29 слов F-мера стала хуже, что, возможно, говорит о трудностях и несовершенстве метода. Для окончательной оценки необходима большая экспериментальная база: не десятки слов, а десятки и сотни тысяч.

Данный метод может быть полезен при различении значений слов без учителя при небольшом количестве обучающих данных.

#### «РАЗРЕШЕНИЕ МНОГОЗНАЧНОСТИ В БИОМЕДИЦИНСКИХ ТЕКСТАХ С ПОМОЩЬЮ МЕТОДОВ КЛАСТЕРИЗАЦИИ БЕЗ УЧИТЕЛЯ»

Е. А. Ярышкина

В статье [27] изучаются уже существующие методы кластеризации без учителя и их эффективность для решения лексической многозначности при обработке текстов по биомедицине. Решение проблем лексической многозначности в данной области включает в себя не только традиционные задачи присвоения ранее определенных смысловых значений для терминов, но так же и обнаружения новых значений для них, ещё не включённых в данную онтологию.



Авторы описали методологию метода решения лексической многозначности без учителя, учитываемые лексические признаки и наборы экспериментальных данных. В качестве оценки эффективности алгоритмов кластеризации текста была предложена F-мера.

Подход для решения поставленной задачи – это разделение контекстов (фрагментов текста), содержащих определенное целевое слово на кластеры, где каждый кластер представляет собой различные значения целевого слова. Каждый кластер состоит из близких по значению контекстов. Задача решается в предположении, что используемое целевое слово в аналогичном контексте будет иметь один и тот же или очень похожий смысл.

Процесс кластеризации продолжается до тех пор, пока не будет найдено предварительно заданное число кластеров. В данной статье выбор шести кластеров основан на том факте, что это больше, чем максимальное число возможных значений любого английского слова, наблюдаемое среди данных (большинство слов имеют два-три значения). Нормализация текста не выполняется.

Данные в этом исследовании состоят из ряда контекстов, которые включают данное целевое слово, где у каждого целевого слова вручную отмечено – какое значение из словаря было использовано в этом контексте. Контекст – это единственный источник информации о целевом слове. Цель исследования – преобразовать контекст в контекстные вектора первого и второго порядка [5]. Контекстные вектора содержат следующие «лексические свойства»: биграммы, совместную встречаемость и совместную встречаемость целевого слова. Биграммами являются как двухсловные словосочетания, так и любые два слова, расположенные рядом в некотором тексте. Для лингвистических исследований могут быть полезны только упорядоченные наборы биграмм [3].

Экспериментальные данные – это набор NLM WSD [31] (NLM – национальная библиотека медицины США), в котором значения слов взяты из UMLS (единая система медицинской терминологии). UMLS имеет три базы знаний:

- Метатезаурис включает все термины из контролируемых словарей (SNOMED-CT, ICD и другие) и понятия, которые представляют собой кластеры из терминов, описывающих один и тот же смысл.
- Семантическая сеть распределяет понятия на 134 категории и показывает отношения между ними. SPECIALIST-

лексикон содержит семантическую информацию для терминов Метатезауриса.

- Medline – главная библиографическая база данных NLM, которая включает приблизительно 13 миллионов ссылок на журнальные статьи в области науки о жизни с уклоном в биомедицинскую область.

Авторы успешно проверили по три конфигурации существующих методов (PB – Pedersen and Bruce [24], SC – Sch?tze [30]) и оценили эффективность использования SVD (сингулярное разложение матриц). Методы PB основаны на контекстных векторах первого порядка – признаки одновременного присутствия целевого слова или биграммы. Рассчитывается среднее расстояние между кластерами или применяется метод бисекций. PB методы подходят для работы с довольно большими наборами данных. Методы SC основаны на представлениях второго порядка – матрицы признаков одновременного присутствия или биграммы, где каждая строка и столбец – вектор признаков первого порядка данного слова. Так же рассчитывается среднее расстояние между кластерами или применяется метод бисекций. SC методы подходят для обработки небольших наборов данных.

Метод SC2 (признаки одновременного присутствия второго порядка, среднее расстояние между элементами кластера в пространстве подобия) с применением и без SVD показал лучшие результаты: всего 56 сравниваемых экземпляров, в 47 случаях метод SC2 показал наилучшие результаты, в 7 случаях результаты незначительно отличаются от других проверяемых методов.

Все эксперименты, указанные в исследовании, выполнялись с помощью пакета SenseClusters [29]. В ходе исследования было проведено два эксперимента для разных наборов данных. Маленький тренировочный набор – это набор NLM WSD, который включает 5000 экземпляров для 50 часто встречаемых неоднозначных терминов из Метатезауриса UMLS. Каждый неоднозначный термин имеет по 100 экземпляров с указанным вручную значением. У 21 термина максимальное число экземпляров находится в пределах от 45 до 79 экземпляров. У 29 терминов число экземпляров от 80 до 100 для конкретного значения. Стоит отметить, что каждый термин имеет категорию «ни одно из вышеупомянутых», которая охватывает все оставшиеся значения, не соответствующие доступным в UMLS. Большой тренировочный набор явля-

ется реконструкцией «1999 Medline», который был разработан Weeber [33]. Были определены все формы из набора NLM WSD и сопоставлены с тезисами «1999 Medline». Для создания тренировочного набора экземпляров использовались только те тезисы из «1999 Medline», которым было найдено соответствие в наборе NLM WSD.

Использование целиком текста аннотации статьи в качестве контекста приводит к лучшим результатам, чем использование отдельных предложений. С одной стороны, большой объем контекста, представленный аннотацией, дает богатую коллекцию признаков, с другой стороны, в коллекции WSD представлено небольшое число контекстов.

## «ВЫЯВЛЕНИЕ ЗНАЧЕНИЙ СЛОВ ИЗ ТЕКСТА»

Д. Ю. Янкевич

В статье [22] представлен алгоритм автоматического обнаружения значений слов в тексте, названный *Кластеризация посредством комитетов* (Clustering By Committee (CBC)). Также авторы предлагают методологию оценки для автоматического измерения точности и полноты найденных значений.

## АЛГОРИТМ

Алгоритм первоначально находит множество небольших кластеров, называемых комитетами, каждый из которых представляет собой одно из значений определяемого слова. Центр тяжести членов комитета (мера связности с определяемым словом) используется в качестве вектора признаков кластера.

CBC состоит из трех этапов.

На этапе I для каждого элемента (слова) вычисляются  $k$  наиболее похожих слов. Сначала весь список относящихся к слову значений сортируется по убыванию значений связи согласно формуле точечной взаимной информации (pointwise mutual information (PMI) [18]), а затем, с помощью иерархического кластерного анализа по *методу средней связи* [4], вычисляется сходство между всеми элементами кластера попарно. Значение функции PMI [18] между предполагаемым значением слова (контекстом) и элементом (словом) вычисляется следующим образом: пусть  $x$  – это рассматриваемый элемент, а  $y$  – контекст. *Точечная взаимная информация* между  $x$  и  $y$  определена как:

$$pmi(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$

При кластеризации посредством метода средних связей (*average-link clustering*) вычисляется среднее сходство между данным объектом и всеми объектами в кластере, а затем, если найденное *среднее значение сходства* достигает или превосходит некоторый заданный пороговый уровень сходства, объект присоединяется к этому кластеру [4]. Сложность этого алгоритма  $O(n^2 * \log(n))$ , где  $n$  – число кластеризуемых элементов [14].

В этапе I оценка (PMI) означает предпочтение большим и тесно связанным кластерам.

На II этапе строится набор небольших кластеров, где элементы каждого кластера образуют комитет. В ходе работы «Алгоритма 1» формируется как можно больше комитетов при условии, что каждый вновь созданный комитет не слишком похож на любой из уже существующих комитетов. Если условие нарушается, комитет просто отбрасывается. Алгоритм описан ниже подробно.

**Исходные параметры:** Список элементов  $E$ , которые будут сгруппированы, база данных сходства  $S$ , из фазы I, пороги  $\theta 1$  и  $\theta 2$  (с помощью порога  $\theta 1$  сохраняются только те кластеры, которые имеют значения, отличные от ранее обнаруженных, порог  $\theta 2$  позволяет обнаружить элементы, не принадлежащие ни одному из кластеров)

**Результат:**  $C$  – список комитетов

**Шаг 1:**

для каждого  $e \in E$  выполнять

1. Кластер  $k$  наиболее “близких” (похожих) элементов  $e$  из  $S$  с помощью метода средней связи
2. Для каждого обнаруженного кластера  $c$  вычислить следующую оценку:  
 $val = |C| * avgsim(c)$ , где  $|C|$  – количество элементов  $c$  и  $avgsim(c)$  – усреднённое сходство между всеми парами элементов кластера  $c$ .
3. Записать кластер  $c$  с наивысшей оценкой в список  $L$

конец цикла

**Шаг 2:**

Сортировка кластеров в списке  $L$  в порядке убывания их оценок  $val$

**Шаг 3:**

$C = \emptyset$  // Пусть перечень комитетов, изначально пустой

для каждого  $c \in L$  выполнять

// в отсортированном по убыванию порядке:

1. Вычислить центр тяжести, усредняя поэлементно значение векторов, и вычислить вектор РМЦ центроида (точно так же, как мы делали для отдельных элементов)
2. Если схожесть  $c$  и центроида каждого комитета, ранее добавленного к  $C$ , ниже порогового  $\theta 1$ , то следует добавить  $c$  в  $C$

конец цикла

**Шаг 4:**

если  $C = \emptyset$  тогда

| вернуть  $C$

конец условия

$R = \emptyset$  //  $R$  – это множество остатков, то есть элементов, не охваченных ни одним из кластеров

для каждого  $e \in E$  выполнять

для каждого  $c \in C$  выполнять

если  $sim(e, c) < \theta 2$  //  $sim(e, c)$  – схожесть  $e$  каждому комитету из  $C$  тогда  
|  $R+ = e$  // то следует добавить  $e$  в список остатков  $R$

конец условия

конец цикла

конец цикла

если  $R = \emptyset$  тогда

| вернуть  $C$  иначе

| вернуть  $C \cup Algorithm1(R, S, \theta 1, \theta 2)$

конец условия

конец условия

### Алгоритм 1: II фаза CBC

В результате второго этапа построения CBC остаются кластеры, связанные более тесно (имеющие большее значение  $val$ , см. Шаг 1 и 3 Алгоритма 1).

На заключительном III этапе работы алгоритма каждый элемент  $e$  присваивается наи-

более подходящему кластеру следующим образом (при этом центроид членов комитета используется в качестве вектора характеристик кластера):

**Результат:** Итоговые кластеры с максимальным значением связи ( $val$ ) между словами  
Пусть список кластеров (изначально пустых)  
Пусть  $S$  это топ-200 кластеров, схожих с  $e$   
**до тех пор, пока  $S$  не пуста выполнять**  
    пусть  $c \in S$  наиболее близкий кластер к  $e$   
    **если сходство ( $e, c$ )  $< \sigma$  тогда**  
        **конец цикла**  
    **конец условия**  
    **если  $c$  не схож ни с одним кластером в  $S$  тогда**  
        присвоить  $e$  к  $c$   
        удалить из  $e$  его характеристики, которые перекрываются с характеристиками  $c$   
    **конец условия**  
удалить  $c$  из  $S$   
**конец цикла**

На III этапе кластер сохраняется только в случае, если его сходство со всеми ранее полученными кластерами ниже установленного порогового значения.

Согласно дистрибутивной гипотезе (Distributional Hypothesis) [12] слова, употребляемые в сходных контекстах, близки по смыслу. Алгоритм *Кластеризации посредством комитетов* [22] разрешает лексическую многозначность, группируя слова согласно сходству их контекстов. Каждому полученному кластеру соответствует одно из значений слова.

В таблице 5 каждая запись показывает кластеры, которым принадлежит заглавное слово. Имена для кластеров Nq34, Nq137, ... генерируются автоматически. После каждого имени кластера находится число, обозначающее сходство между кластером и заглавным словом (т.е. рукав, сердце и одежда). Далее перечисляются четыре слова, наиболее близкие центроиду кластера. Каждый кластер соответствует одному значению заглавного слова. Например, Nq34 соответствует значению «деталь одежды», а Nq137 соответствует значению «ответвление русла реки».

Таблица 5. Для построения кластеров использовались данные словарных статей Викисловаря: «одежда», «рукав» и «сердце»

(Рукав	
Nq34 0.39	(деталь, манжета, полотно)
Nq137 0.20	(протока, русло, отмель, поток)
Nq217 0.18	(шланг, труба, огнетушитель)
)	
(Сердце	
Nq72 0.27	(орган, костный мозг, почка)
Nq866 0.17	(душа, рассудок, сознание)
)	
(Одежда	
Nq215 0.41	(мануфактура, юбка, брюки)
Nq235 0.20	(покрытие, оболочка, дорога)
)	

## СРАВНЕНИЕ С АЛГОРИТМОМ UNICON

СВС является разновидностью алгоритма UNICON [17], который также строит центроид кластера, используя небольшой набор похожих элементов.

Одним из основных различий между UNICON и СВС является то, что UNICON гарантирует, что различные комитеты не имеют одинаковых элементов, тем не менее, центры тяжести двух комитетов по-прежнему могут быть очень близкими (похожими). В UNICON'e эта проблема решается объединением таких кластеров. В отличие от этого, на II этапе СВС создаются только те комитеты, центры тяжести которых отличны от всех ранее созданных комитетов.

Есть разница и на III этапе СВС. Алгоритм UNICON плохо работает со словами, которые имеют несколько широко используемых (доминирующих) значений. Например, пусть значение «отмычка» является более употребимым для слова «ключ», чем значение «водный источник». Приведём смесь слов-синонимов к разным значениям слова «ключ»: пневмоключ, электроключ, родник, родничок, источник, криница, гидроключ, ключик, тангента, треншальтер, тумблер, знак, контролька, отпирка, виброплекс, шифр. В этом списке 10 значений относятся к значению «отмычка()», 4 к «водный источник()» и 2 к значению «знак()». По этому списку алгоритмом UNICON будут сгенерированы кластеры «отмычка», «шифрование», «происхождение», «криптография», «кнопка», «водный источник», «переключатель», «намёк». Сходство между словом и полученными кластерами является очень низким, к тому же есть кластеры, содержащие одинаковые слова. С другой стороны, СВС удаляет «пересекающиеся» (общие для двух кластеров) характеристики после того, как присвоит значение кластеру (допустим характеристики, относящиеся к значению «отмычка» слова «ключ» из вектора характеристик «отмычка»). В ре-

зультате, сходство между кластером “водный источник” {родник, родничок, источник, кри-ница} и пересмотренным вектором характери-стик кластера “водный источник” становится намного выше. Что в свою очередь приводит к тому, что кластеры становятся гораздо точ-нее.

## WSD-методы, основанные на зна-ниях

### «ПОСТРОЕНИЕ СОЧЕТАЕМОСТНЫХ ОГРА-НИЧЕНИЙ НА ОСНОВЕ БАЙЕСОВСКИХ СЕ-ТЕЙ ДЛЯ РАЗРЕШЕНИЯ МНОГОЗНАЧНО-СТИ»

И. А. Сихонина

В статье [9] представлена Байесовская мо-дель, применяемая для разрешения лексиче-ской многозначности глаголов. Авторы рас-сматривают такое понятие, как сочетаемост-ные ограничения (selectional preferences). *Со-четаемостные ограничения* (далее SP) – это закономерности использования глагола отно-сительно семантического класса его парамет-ров (субъект, объект (прямое дополнение) и косвенное дополнение).

Модели автоматического построения SP важны сами по себе и имеют приложения в об-работке естественного языка. Сочетаемостные ограничения глагола могут применяться для получения возможных значений неизвестно-го параметра при известных глаголах; напри-мер, из предложения «Осенние *хххх* жу-жали и бились на стекле» легко определить, что “хххх” – мухи. При построении предложе-ния SP позволяют отранжировать варианты и выбрать лучший среди них. Исследование SP могло бы помочь в понимании структуры мен-тального лексикона.

Системы обучения SP без учителя обычно комбинируют статистические подходы и под-ходы, основанные на знаниях. Компонент ба-зы знаний (здесь WordNet [19]) – это обычно база данных, в которой слова сгруппированы в классы.

Статистический компонент состоит из пар предикат-аргумент, извлечённых из неразмеченного корпуса. В тривиальном алгоритме можно было бы получить список слов (пря-мых дополнений глагола), и для тех слов, ко-торые есть в WordNet, вывести их семантиче-ские классы. В работе [9] семантическим клас-сом называется синсет (группа синонимов) те-зауруса WordNet, то есть класс соответству-ет одному из значений слова. Таким обра-

зом, в тривиальном алгоритме на основе дан-ных WordNet можно выбрать классы (значе-ния слов), с которыми употребляются (встре-чаются в корпусе) глаголы.

Например, если в исходном корпусе текстов глагол *ползатъ* употребляется со словом *яще-рица*, принадлежащим классу РЕПТИЛИИ, то в модели построения SP будет записано, что «глагол *ползатъ* употребляется со слова-ми из класса РЕПТИЛИИ». Если слово *кро-кодил*, во-первых, также встречается в тексте с глаголом *ползатъ*, во-вторых, слово *кроко-дил* принадлежит сразу двум классам: РЕП-ТИЛИЯ и ВЕРТОЛЁТ, то из этого следует, что модель SP будет расширена информацией о том, что «глагол *ползатъ* употребляется со словами из классов и РЕПТИЛИЯ, и ВЕРТО-ЛЁТ».

В ранее разработанных моделях (Резник (1997) [26], Абни и Лайт (1999) [6]) было обна-ружено, что главная трудность в таком триви-альном алгоритме – это наличие неоднознач-ных слов в обучающих данных. В тех же ра-ботах ([26], [6]) были предложены более слож-ные модели, в которых предполагается, что все значения многозначных слов появляются с одинаковой частотой.

**Байесовские сети** или Байесовские сети доверия (БСД) состоят из множества перемен-ных (вершин) и множества ориентированных ребер, соединяющих эти переменные. Такой сети соответствует ориентированный ациклич-ный граф. Каждая переменная может прини-мать одно из конечного числа взаимоисклю-чающих состояний. Пусть все переменные бу-дут бинарного типа, то есть принимают од-но из двух значений: истина или ложь. Лю-бой переменной  $A$  с родителями  $B_1, \dots, B_n$  соответствует таблица условных вероятностей (conditional probability table, далее CPT).

Например, построим SP для глагола *пол-затъ* и сеть на рисунке 5 будет базой знаний.

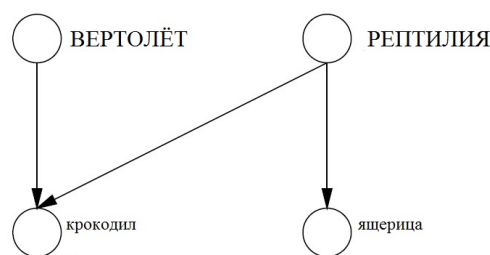


Рис. 5. Байесовская сеть для многозначного суще-ствительного *крокодил*

Глагол *ползать* употребляется со словами *крокодил* и *ящерица*. Переменные ВЕРТОЛЁТ и РЕПТИЛИЯ соответствуют более общим абстрактным значениям, переменные *крокодил* и *ящерица* являются более узкими, конкретными значениями. Переменная РЕПТИЛИЯ может принимать одно из двух значений, соответствующих словам *крокодил* и *ящерица*, именно эту задачу определения значения и нужно решить.

Таблица 6. Условные вероятности переменных *крокодил* и *ящерица* в зависимости от значений переменных ВЕРТОЛЁТ и РЕПТИЛИЯ, где (В, Р, к, я – это аббревиатуры слов ВЕРТОЛЁТ, РЕПТИЛИЯ, *крокодил* и *ящерица*)

	$P(X = x   Y_1 = y_1, Y_2 = y_2)$			
	В,Р	В,¬Р	¬В,Р	¬В,¬Р
к = true	0,99	0,99	0,99	0,01
к = false	0,01	0,01	0,01	0,99
я = true	0,99	0,99	0,01	0,01
я = false	0,01	0,01	0,99	0,99

При построении таблицы 6 условных вероятностей (СРТ), учтём следующие предположения:

- вероятность, что выбираем какой-либо из концептов (ВЕРТОЛЁТ и РЕПТИЛИЯ) очень мала, то есть  $P(B=true) = P(P=true) = 0,01$ , следовательно, велика вероятность, что концепты не выбраны:  $P(B=false) = P(P=false) = 0,99$ ;
- если какой-либо из концептов истинен (В, Р), то «выпадает» слово *крокодил*;
- если концепт РЕПТИЛИЯ истинен, то растут шансы встретить слово *ящерица*;

Из таблицы 6 вероятности появления слов следует вывод, что использование разу двух значений слова *крокодил* (*рептилия* и *вертолёт МИ-24*) маловероятно. Вероятность использования значения РЕПТИЛИЯ намного больше чем значения ВЕРТОЛЁТ. Таким образом гипотеза «вертолёт» «отброшена» (“explaining away”).

**Байесовские сети для построения SP.** Иерархия существительных в WordNet представлена в виде ориентированного ациклического графа. Синсет узла принимает значение «истина», если глагол «выбирает» существительное из набора синонимов. Априорные вероятности задаются на основе двух предположений: во-первых, маловероятно, что

глагол будет употребляться только со словами какого-то конкретного синсета, и во-вторых, если глагол действительно употребляется только со словами из данного синсета (например, синсет ЕДА), тогда должно быть правомерным употребление этого глагола с гипонимами этого синсета (например, ФРУКТ).

Те же предположения (что для синсетов) верны и для употреблений слов с глаголами:

1. слово, вероятно, является аргументом глагола в том случае, если глагол употребляется с каким-либо из значений этого слова;
2. отсутствие связки глагол-синсет говорит о малой вероятности того, что слова этого синсета употребляются с глаголом;

Словам «вероятно» и «маловероятно» должны быть приписаны такие числа, сумма которых равна единице.

Находкой работы [9] является разъяснение стратегии “explaining away”, то есть отбрасывание маловероятных значений слов при построении сочетаемостных ограничений. Такая стратегия является неотъемлемым свойством Байесовских сетей и Байесовского вывода, полезным свойством при разрешении лексической многозначности.

## Библиография

Библиографические ссылки принято оформлять в виде [номер], в отличие от ранее принятых [Автор, год] (см., напр., [1]). Источник, процитированный выше, был набран командой

```
\bibitem{Trans}
\textit{Борисов~Г.~А.,
Тихомирова~Т.~А.}
Характеристики и свойства потерь
энергии и мощности на пределах
энергетического хозяйства региона //
Труды Карельского научного центра
Российской академии наук. 2010.
\No 3. С.~4--10.
```

## ТЕОРЕМОПОДОБНЫЕ ОКРУЖЕНИЯ

Для теорем, утверждений и пр. необходимо использовать соответствующие окружения. Например:

**Утверждение 1.** В предложенной модели системы обслуживания при  $\rho = ES/ET < 1$  условие  $ES^{\alpha+1} < \infty$  является достаточным для конечности момента порядка  $\alpha$  времени ожидания в системе,  $ED^{\alpha} < \infty$ .

*Доказательство.* Очевидно. □

В данном случае было использовано окружение `\begin{State}...\end{State}`. Для набора доказательства использовалось окружение `\begin{proof}...\end{proof}`. Доступные автору теоремоподобные окружения перечислены в Таблице 7.

Таблица 7. Теоремоподобные окружения

Theorem	Теорема
Lemma	Лемма
State	Утверждение
Corollary	Следствие
Axiom	Аксиома
Definition	Определение
Example	Пример
Remark	Замечание

Для определений, примеров и замечаний используется прямое написание.

**Пример.** Например, как в этом примере.

Соответствующие версии окружений «со звездой» также работают. Пример выше был набран такой командой:

```
\begin{Example*}
Например, как в этом примере.
\end{Example*}
```

Таблица 8. Таблица, демонстрирующая возможность размещения на всю ширину страницы

Первая колонка	вторая колонка	третья колонка	четвертая колонка	пятая колонка
----------------	----------------	----------------	-------------------	---------------

## Рисунки и таблицы

Рисунки и таблицы могут вставляться как на всю ширину страницы, так и на ширину колонки. Желательно использовать рисунки формата pdf. Для конвертации из формата eps можно воспользоваться утилитой `epstopdf`. Так, например, Рис. 6 был вставлен на ширину колонки командой

delay\_80.pdf

Рис. 6. Задержки в модели 10-узловой системы

```
\begin{figure}[H]
\includegraphics[keepaspectratio=true,
width=0.9\columnwidth]{delay_80.pdf}
\caption{Задержки в модели
10-узловой системы}
\label{fig1}
\end{figure}
```

## РАЗМЕЩЕНИЕ НА ВСЮ ШИРИНУ СТРАНИЦЫ

В стилевом файле предусмотрена возможность размещения формул, рисунков и таблиц на ширину страницы. Для этого размещаемый элемент необходимо заключить между командами `\bfullwidth` и `\efullwidth`.

Пример формулы на всю ширину страницы:

$$Y = A_1x + A_2x^2 + \dots + A_nx^n. \quad (1)$$

Набран пример следующим образом:

```
\bfullwidth
\begin{equation}
Y=A_1x+A_2x^2+\ldots +A_nx^n.
\end{equation}
\efullwidth
```

Пример размещения таблицы на ширину страницы (см. таблицу 8):

```
\bfullwidth
\centering
\begin{table}[H]
\begin{tabular}{|c|c|c|c|c|}
\hline
Первая колонка & ...\\
\hline
\end{tabular}
\label{tab_width}
\end{table}
\efullwidth
```

Рис. 1 демонстрирует возможности вставки по всей ширине страницы. Это было достигнуто при помощи команды

```
\bfullwidth
\begin{figure}
\includegraphics[height=100mm,...]
\caption{Задержки в модели...}
\label{fig5}
\end{figure}
\efullwidth
```

Следует обратить внимание на то, что вышеуказанная команда вставит рисунок не ближе, чем на следующей странице сверху, а не сразу на месте указания команды.

## СВЕДЕНИЯ ОБ АВТОРАХ

После основного текста оформляются сведения об авторах. Используется окружение `\begin{aboutauthors}... \end{aboutauthors}`. При этом следует обратить внимание, что работа ведется в двухколоночном режиме, поэтому необходимо вручную указать разрыв колонки для отделения сведений на русском и английском языках. Например:

```
\begin{aboutauthors}
\authorsname{Румянцев Александр...}
аспирант\\
...
\columnbreak
\authorsname{Rumyantsev, Alexander}
...
\end{aboutauthors}
```

## ЗАКЛЮЧЕНИЕ

Компиляцию исходного файла желательно выполнять с помощью макроса `pdflatex`.

В работе рассмотрены основные технические аспекты подготовки статьи для сборника Трудов Карельского научного центра РАН. Предложения и пожелания по доработке стилевого файла, а также текста этого документа принимаются по электронному адресу, указанному в разделе «Сведения об авторах».

## ЛИТЕРАТУРА

1. Борисов Г. А., Тихомирова Т. А. Характеристики и свойства потерь энергии и мощности на пределах энергетического хозяйства региона // Труды Карельского научного центра Российской академии наук. 2010. №3. С. 4–10.
2. Львовский С. М. Набор и верстка в системе L<sup>A</sup>T<sub>E</sub>X. М., 2003. 448 с.
3. Аверин, А.Н. Разработка сервиса поиска биграмм // Труды международной конференции «Корпусная лингвистика–2006. СПб., С.Петербург. ун-та., 2006.
4. Ким Дж. О., Мьюллер Ч.У., Клекка У.Р. 1989. Факторный, дискриминантный и кластерный анализ. Стр.172. «Финансы и статистика», Москва, Россия.
5. Ендрев, А. С. Применение контекстных векторов в классификации текстовых документов. 2010. <http://jre.cplire.ru/iso/oct10/1/text.html>
6. Abney, S. and Light, M. Hiding a semantic hierarchy in a markov model. In Proceedings of the Workshop on Unsupervised Learning in Natural Language Processing, ACL. 1999.
7. M. Berry, T. Do, G. O'Brien, V. Krishna, and S. Varadhan SVDPACK (version 1.0) user's guide. Technical Report CS-93-194, University of Tennessee at Knoxville, Computer Science Department, April 1993.
8. Bruce, R. and Wiebe, J. Word-sense disambiguation using decomposable models. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 1994, pages 139–146.
9. Ciaramita, M. and Johnson, M. Explaining away ambiguity: Learning verb selectional preference with Bayesian networks. 2000.
10. COTTRELL, G. W. and SMALL, S. L. A connectionist scheme for modelling word sense disambiguation – Cognition and brain theory. 1983. № 6. P. 89–120.
11. Charles X. Ling, Marin Marinov Answering the connectionist challenge: A symbolic model of learning the past tenses of English verbs, Cognition, Elsevier, 1993



12. *Harris, Z.* Distributional structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics*. New York: Oxford University Press. 1985. pp. 26–47
13. *Jain, A. and Dubes, R.* *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1988.
14. *A. Jain, M. Murthy, and P. Flynn* Data clustering: a review. *ACM Computing Surveys*, 31(3):264-323, September 1999.
15. *C. Leacock, G. Towell, and E. Voorhees.* Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 260–265, March. 1993.
16. *LESK, M.* Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone – *Proceedings of the 5th SIGDOC*. New York. 1986. P. 24–26.
17. *Lin, D. and Pantel, P* 2001. Induction of semantic classes from natural language text. In *Proceedings of SIGKDD-01*. pp. 317–322. San Francisco, CA
18. *Manning, C. D. and Schütze, H.* 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
19. *Miller, G.* Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4). 1990.
20. *Raymond J. Mooney* Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning, Department of Computer Sciences, University of Texas, Austin, TX 78712-1188, 1996
21. *R. J. Mooney, M. E. Califf* Induction of First-Order Decision Lists: Results on Learning the Past Tense of English Verbs, Department of Computer Sciences, University of Texas, Austin, TX 78712-1188, 1995
22. *Patrick Pantel, Dekang Lin* Discovering Word Senses from Text. University of Alberta. Department of Computing Science Edmonton, Alberta T6H 2E1 Canada, 2002.
23. *Pedersen, T.* A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation // Department of Computer Science, University of Minnesota Duluth. – 2000.
24. *Pedersen, T. and Bruce, R.* Distinguishing word senses in untagged text. *Proc. EMNLP*. Providence, RI, 1997.
25. *Purandare, A. and Pedersen, T.* Improving word sense discrimination with gloss augmented feature vectors // *Workshop on Lexical Resources for the Web and Word Sense Disambiguation*. – 2004. – C. 123-130.
26. *Resnik, P.* Selectional preference and sense disambiguation. In *Proceedings of the ANLP-97 Workshop: Tagging Text with Lexical Semantics: Why, What, and How?* 1997.
27. *Savova, G.* Resolving ambiguities in biomedical text with unsupervised clustering approaches. University of Minnesota Supercomputing Institute Research Report, 2005.
28. *VERONIS, J. and IDE, N.* Word sense disambiguation with very large neural networks extracted from machine readable dictionaries – *Proceedings of the 13th International Conference on Computational Linguistics*. Helsinki. 1990. P. 389–394.
29. *SenseClusters* <http://senseclusters.sourceforge.net>
30. *Schütze, H.* Automatic Word Sense Discrimination. *Computational Linguistics*, vol. 24, number 1., 1998.
31. *UMLS Terminology Services (UTS)*. <http://umlsk.nlm.nih.gov/kss/servlet/Turbine/template>
32. *WALTZ, D. L. and POLLACK, J. B.* Massively parallel parsing: a strongly interactive model of natural language interpretation – *Cognitive science*. 1985. № 9. P. 51–74.
33. *Weeber, M. and Mork, J. and Aronson, A.* Developing a test collection for biomedical word sense disambiguation. *Proc. AMIA.*, 2001.
34. *Zhao, Y. and Karypis, G.* Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 515-524, McLean, VA, 2002.

## СВЕДЕНИЯ ОБ АВТОРЕ:

### Кирилов Александр Николаевич

доктор физико-математических наук  
доцент

Институт прикладных математических исследований  
КарНЦ РАН

ул. Пушкинская, 11, Петрозаводск, Республика Карелия, Россия, 185910

эл. почта: kirillov@krc.karelia.ru

тел.: (8142) 766312

### Kirilov, Alexander

Doctor (DSc) of Physics and Mathematics  
Assistant Professor

Institute of Applied Mathematical Research, Karelian  
Research Centre, Russian Academy of Science

11 Pushkinskaya St., 185910 Petrozavodsk, Karelia,  
Russia

e-mail: kirillov@krc.karelia.ru

tel.: (8142) 766312

**Коржицкий Никита Иванович**

Студент  
Математический факультет  
Петрозаводский государственный университет  
пр-кт Ленина, 33, Петрозаводск, Республика Карелия  
тел.: +7(8142) 71-10-78  
nikita@nikita.tv

**Румянцев Александр Сергеевич**

аспирант  
Институт прикладных математических исследований  
КарНЦ РАН  
ул. Пушкинская, 11, Петрозаводск, Республика Карелия, Россия, 185910  
эл. почта: ar0@krc.karelia.ru  
тел.: (8142) 763370

**Сихонина Ирина Александровна**

Студентка  
Математический факультет  
Петрозаводский государственный университет  
пр-кт Ленина, 33, Петрозаводск, Республика Карелия  
тел.: +7(8142) 71-10-78  
syawenka@mail.ru

**Спиркова Анна Михайловна**

Студентка  
Математический факультет  
Петрозаводский государственный университет  
пр-кт Ленина, 33, Петрозаводск, Республика Карелия  
тел.: +7(8142) 71-10-78  
annspirkova@gmail.com

**Ткач Станислав Сергеевич**

Студент  
Математический факультет  
Петрозаводский государственный университет  
пр-кт Ленина, 33, Петрозаводск, Республика Карелия  
тел.: +7 (8142) 71-10-78  
tkachkras@gmail.com

**Ярышкина Екатерина Александровна**

Студентка  
Математический факультет  
Петрозаводский государственный университет  
пр-кт Ленина, 33, Петрозаводск, Республика Карелия  
+7 (8142) 71-10-78  
kate.rysh@gmail.com

**Korzhitsky, Nikita**

Student  
Faculty of Mathematics  
Petrozavodsk State University  
Prospect Lenina, 33, Petrozavodsk, Republic of Karelia  
tel.: +7(8142) 71-10-78  
nikita@nikita.tv

**Rumyantsev, Alexander**

Institute of Applied Mathematical Research, Karelian  
Research Centre, Russian Academy of Science  
11 Pushkinskaya St., 185910 Petrozavodsk, Karelia,  
Russia  
e-mail: ar0@krc.karelia.ru  
tel.: (8142) 763370

**Sikhonina, Irina**

Student  
Faculty of Mathematics  
Petrozavodsk State University  
Prospect Lenina, 33, Petrozavodsk, Republic of Karelia  
tel.: +7(8142) 71-10-78  
syawenka@mail.ru

**Spirikova, Anna**

Student  
Faculty of Mathematics  
Petrozavodsk State University  
Prospect Lenina, 33, Petrozavodsk, Republic of Karelia  
tel.: +7(8142) 71-10-78  
annspirkova@gmail.com

**Tkach, Stanislav**

Student  
Faculty of Mathematics  
Petrozavodsk State University  
Prospect Lenina, 33, Petrozavodsk, Republic of Karelia  
tel.: +7 (8142) 71-10-78  
tkachkras@gmail.com

**Yaryshkina, Ekaterina**

Student  
Faculty of Mathematics  
Petrozavodsk State University  
Prospect Lenina, 33, Petrozavodsk, Republic of Karelia  
+7 (8142) 71-10-78  
kate.rysh@gmail.com