АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

81'374:004.82

А. Ф. Гельбух (Мексика), Г. О. Сидоров (Мексика),
Санг-Ёнг Хан (Южная Корея)

Автоматическое разрешение неоднозначности значений слов в словарных толкованиях

Приводится метод автоматического разрешения неоднозначности значений слов в словарных статьях толковых словарей и описывается его применение к толковому словарю испанского языка. Значения слов берутся из того же самого словаря. Для определения наиболее вероятного значения слова, используемого в толковании, применяется улучшенный алгоритм Леска. В отличие от известного алгоритма Леска, вероятность каждого значения вычисляется с учетом нескольких факторов. Так, при сравнении определений с контекстом используются синонимы, словообразовательные дериваты, а также слова, входящие в толкования слов, входящих в толкование данного слова. По сравнению с общей задачей разрешения неоднозначности значений слов в произвольных текстах, ее сужение на словарные толкования позволяет упростить алгоритм — например, не нужно вычислять размер окна контекста, используемого для определения весов.

1. ВВЕДЕНИЕ

Для описания значений слов в любом толковом словаре используются другие слова того же языка. Например, слово *кошка* может толковаться как *домашнее животное из породы кошачьих*. Казалось бы, словарная статья устанавливает отношение между заглавным словом (*кошка*) и словами, входящими в толкование (*домашнее, животное, из, породы, кошачьих*). Однако на самом деле словарь задает такое отношение не между словами, а между значениями слов, с одной стороны (например, *кошка*$_1$ — *домашнее животное из породы кошачьих*, а *кошка*$_2$ — *предмет альпинистского снаряжения*), и буквенными цепочками (*домашнее, породы*) — с другой. В языке же соответствующее отношение существует между конкретными значениями слов: например, в определении *кошка*$_1$ цепочка *порода* должна означать *порода*$_1$ — *порода животных*, а не *порода*$_2$ — *горная порода*.

Хотя человек обычно без труда определяет нужное значение, для применения словарей в автоматических естественно-языковых системах, такой выбор без участия человека представляет серьезную проблему. В настоящей статье мы обсуждаем алгоритмы, помогающие компьютеру выбрать наиболее вероятные значения слов, употребляемых в словарных толкованиях. В литературе такая задача называется задачей разрешения неоднозначности значений слов (WSD — word sense disambiguation), при этом подразумевается, что выбор делается автоматически, без участия человека.

Важность такой проблемы достаточно очевидна: ни одна программа, связанная с содержательной — принимающей во внимание смысл текста — автоматической обработкой естественного языка, не может работать с необходимым уровнем надежности, если она не может правильно выбирать значения слов. Например, при информационном поиске для правильного ответа на запрос "конструкция кошек для гранитных склонов" поисковая система должна (автоматически) отличать вхождения значения *кошка*$_1$ от вхождений значения *кошка*$_2$ во всех документах базы. Программа автоматического перевода должна перевести упомянутый выше запрос на английский язык как "*design of grapplers for granite slopes*", а не как "*design of *cats for granite slopes*". В качестве еще одного примера можно упомянуть трансформацию толкового словаря в тезаурус, используемый в задачах искусственного интеллекта [1].

Актуальность проблемы разрешения неоднозначности значений слов демонстрирует, например, прошедшее в июле 2001 г. мировое первенство систем автоматического разрешения неоднозначности значений слов Senseval-2*, в котором участвовали 94 системы, созданные в 35 различных научных организациях. Соревнование проводилось на текстах на 12 языках (английский, баскский, голландский, датский, испанский, итальянский, китайский, корейский, чешский, шведский, эстонский, японский). Значения слов брались из соответствующих вариантов словаря WordNet, что позволяло сравнивать результаты, полученные различными системами, несмотря на то, что версии этого словаря для всех языков, кроме английского, оставляют желать лучшего.

Необходимо отметить, что большинство систем, представленных на Senseval-2, использовало статистические методы. Из методов, основанных на знаниях, ни одна из систем не использовала алгоритм, основанный на методе Леска, разъясняемом в следующем параграфе. Тем не менее, мы полагаем, что этот метод может быть существенно улучшен и успешно применен на практике.

Далее в статье мы кратко обсуждаем существующие подходы к решению задачи разрешения неоднозначности значений слов, описываем предлагаемый нами алгоритм, основанный на идее метода Леска, и обсуждаем результаты наших экспериментов с толковым словарем испанского языка.

2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

Проблема автоматического разрешения неоднозначности значений слов имеет достаточно богатую историю. Существуют два основных подхода

* www.sle.sharp.co.uk/senseval2.

к этой проблеме, один из которых можно назвать основанным на статистике, а второй — на знаниях. На данном этапе преобладающим является подход, основанный на статистике, т. е. использующий исключительно статистические методы работы с корпусом текстов, без привлечения дополнительных источников информации о языке [2, 3]. К широко применяемым методам относятся, например, байесовские классификаторы, метод разделяющего вектора (support vector machine) и другие чисто статистические методы. В рамках такого подхода обычно требуется предварительное обучение системы, и, как следствие, ручная разметка больших массивов данных.

Однако в последнее время снова пробуждается интерес и ко второму типу подходов — подходам, основанным на знаниях. При таком подходе привлекаются достаточно большие дополнительные источники лингвистической информации, например, словари различных типов. Исторически первыми методами разрешения неоднозначности значений слов — как и в компьютерной лингвистике в целом — были методы, развиваемые в рамках именно этого подхода, как, например, методы, которые предложили еще Lesk [4] и Hirst [5]. Преимущество подхода, основанного на привлечении дополнительных источников знаний, состоит в его прозрачности, т. е. система может шаг за шагом дать объяснение своих решений. Но все больше существующих и не требующих специальной разработки лингвистических ресурсов (словарей, корпусов и т. п.) становятся доступными научному сообществу, занимающемуся автоматической обработкой естественного языка, что существенно расширяет возможности подхода, основанного на знаниях.

Для разрешения неоднозначности слов Lesk [4] предложил следующий алгоритм. Для каждого значения рассматриваемого слова подсчитывается число слов, упомянутых как в словарном определении данного значения, так и в ближайшем контексте рассматриваемого вхождения слова. В качестве наиболее вероятного значения выбирается то, для которого такое пересечение оказалось больше. В качестве слов Леск рассматривал буквенные цепочки, что оправданно для английского языка.

Например, рассмотрим определения, приведенные в предыдущем параграфе:

(1) *кошка*$_1$ — *домашнее животное из породы кошачьих,*
(2) *кошка*$_2$ — *предмет альпинистского снаряжения*

и текст *в Китае выведены новые породы кошек.* С определением 1) у этого текста одно общее слово — *породы*, а с определением 2) — ни одного. Следовательно, алгоритм Леска выберет значение *кошка*$_1$.

В последнее время появилось большое количество работ, предлагающих использовать модификации алгоритма Леска. В этих работах выдвигаются идеи, связанные с дополнительным использованием различных словарей (тезаурусы, словари синонимов) или моделей (морфологические, синтаксические и т. п.) [6–13]. Необходимо заметить, что все эти работы, кроме [7], ориентированы на обработку обычных текстов, а не словарей, и ни одна не использует в качестве материала для обработки именно толковый словарь. Кроме того, практически всегда дело ограничивается достаточно небольшими экспериментами и не производится обработка достаточно больших массивов данных.

3. АЛГОРИТМ

В качестве возможных путей улучшения исходного алгоритма Леска очевидным образом напрашивается, во-первых, привлечение дополнительной информации о сходстве слов и, во-вторых, учет различной значимости совпадения для разных слов. Нами был разработан улучшенный вариант алгоритма, где в качестве дополнительной информации используются словарь синонимов, словообразовательная морфологическая модель, а также привлекаются толкования слов, входящих в исходное толкование.

Важным моментом в настоящей работе является то, что алгоритм разрешения неоднозначности значений слов применяется к толкованиям, берущимся из словаря, что существенно упрощает задачу по сравнению с применением алгоритма к обычным текстам по следующим причинам:

все слова толкования заведомо связаны с заглавным словом, поскольку входят в его определение;

следовательно, не возникает проблема выбора размера окна контекста, в котором надо рассматривать слова, а используется все определение целиком;

разрешение неоднозначности частей речи (что обычно является первым шагом подобных алгоритмов) упрощено, поскольку толкования являются структурированными и, следовательно, части речи слов на определенных местах предсказуемы; кроме того, помогает информация о грамматическом классе заглавного слова.

Напомним стоящую перед нами задачу. Для каждого слова, входящего в толкование какого-либо слова, рассматриваются его собственные толкования из этого же словаря. Проблема состоит в выборе самого подходящего из нескольких значений данного слова. Например, для определения

(3) *Вискас — вид корма для кошек и других домашних животных*

проблема состоит в (автоматическом) выборе значения (1) или (2) для слова *кошка.*

Наш алгоритм работает в два этапа: предобработка и вычисление весов, выражающих вероятности различных значений рассматриваемого слова. В результате выбирается значение слова, имеющее максимальный вес. В случае, если несколько значений слова имеют равные веса, то берется первое из таких значений, поскольку, как показывают наши экспериментальные данные, обычно лексикографы помещают интуитивно более частотные значения первыми (см. п. 4.1).

3.1. Предобработка

Целью предобработки является лемматизация (приведение словоформ к лемме, т. е. к нормальному — словарному — виду: инфинитив для глагола, именительный падеж единственного числа для существительного и т. д.) и разрешение неоднозначности частей речи (part-of-speech tagging).

В наших экспериментах с испанским словарем для лемматизации использовалась система морфологического анализа испанского языка, разработанная в нашей лаборатории. В данный момент в словаре системы около 100.000 основ, что позволяет распознавать около 500.000 словоформ. Для разрешения омонимии частей речи были разработаны синтаксические эвристики для испанского языка, похожие на правила синтаксического и предсинтаксического анализа системы ЭТАП-1 [14]. Некоторые эвристики связаны с синтаксической структурой предложений: например, слово, перед которым стоит артикль (кроме el), не может быть глаголом. Другие эвристики основаны на информации о структуре толкований, например, первое слово толкования обычно имеет ту же часть речи, что и заглавное слово.

Другая важная часть предобработки — удаление из статей слов, относящихся к служебным частям речи (предлогов, союзов, вспомогательных глаголов и т. д.). Это необходимо, потому что эти слова не вносят дополнительной лексической информации, зато могут внести нежелательный шум.

Таким образом, после предобработки толкование каждого слова сводится к набору лемм значащих слов с однозначно приписанными частями речи, например:

(1а) *кошка*$_1$ — *домашний*$_{\text{прил}}$ *животное*$_{\text{сущ}}$ *порода*$_{\text{сущ}}$ *кошачий*$_{\text{прил}}$.

В дальнейшем, говоря о словах в толковании, будем иметь в виду соответствующие словам символы (лемма и часть речи) в таком обработанном толковании. Под толкованием будем понимать множество, состоящее из таких "слов".

3.2. Вычисление весов значений слов

Рассмотрим слово W, входящее в некоторое толкование S слова H (по определению, заглавное слово не является частью толкования). Предположим, что для слова W в словаре найдено несколько возможных значений $s_1, \ldots, s_n$. Каждому значению s_i в свою очередь соответствует множество слов — его толкование. В качестве веса значения вхождения слова s_i в данном толковании S будем использовать меру близости между множеством s_i и множеством S без самого слова W (см. ниже), но с заглавным словом H.

Мера близости определяется следующим образом. Пусть A и B — два множества слов, тогда мера близости $w(A,B) = \sum_{x\in A,\, y\in B} w(x,y)$, где $w(x,y)$ — мера близости слов из множества A и B, соответственно, вычисляемая по следующим правилам:

1. Если слова x и y совпадают, то $w(x,y) = 1,0$.
2. Иначе, одно из слов является синонимом другого, тогда $w(x,y) = 0,5$.
3. Иначе, одно из слов является морфологическим дериватом другого, тогда $w(x,y) = 0,5$.
4. В противном случае — $w(x,y) = 0$.
5. Дополнительно, если одно из слов содержится в толковании хотя бы одного значения другого слова, то значение $w(x,y)$ увеличивается на 0,1.

Константы 1,0, 0,5 и 0,1, фигурирующие в данных правилах, выбраны эмпирически. В дальнейшем мы планируем провести эксперименты по выбору оптимальных значений этих параметров.

Как было сказано, само слово W удаляется из рассматриваемого толкования S, чтобы не сравнивать его с его же значениями. Последнее внесло бы в веса значений нежелательный постоянный компонент, не зависящий от контекста, дающий неоправданное преимущество некоторым из значений — скажем, содержащим само слово W, его дериваты или синонимы.

Введение в рассмотрение толкований слов, содержащихся в исходном толковании (п. 5), но при этом с малым весом, призвано облегчить выбор наилучшего значения в случае равных величин, полученных с учетом остальных факторов. При этом мы опираемся на содержательные данные, а не на вероятностные соображения, например, о том, что более частотные значения обычно ставятся первыми (см. п. 4.1). Если и с учетом толкований веса все-таки окажутся равными, то для принятия решения используются указанные вероятностные соображения. Толкования слов могут браться (как это и было в наших экспериментах) из того же самого толкового словаря, к которому применяется алгоритм.

3.3. Схема работы алгоритма

Итак, приведем схему работы алгоритма в целом. Введем обозначения для следующих функций:

Лемма (x) — первая (словарная) форма буквенной цепочки x, полученная в результате обращения к морфологическому анализатору, например, инфинитив для глагола, единственное число мужского рода именительного падежа для прилагательного и т. д. Вычисление этой функции подразумевает разрешение морфологической неоднозначности.

Синонимы (x) — множество синонимов слова x, т. е. соответствующая словарная статья из внешнего словаря синонимов. Функция вычисляется путем обращения к базе данных словаря синонимов.

Дериваты (x) — множество морфологических дериватов слова x. Функция вычисляется путем обращения к морфологической подсистеме.

Статья (x)$= \bigcup_{i=1}^{n} s_i$ — множество всех слов, входящих в определение хотя бы одного смысла слова x. Если слово не имеет толкования в словаре, то такое множество пусто.

Тогда основную идею алгоритма можно выразить следующим образом:

Дано: некое словарное толкование S слова H и слово $W \in S \setminus \{H\}$, имеющее в данном словаре толкования $s_1, \ldots, s_n$.

Найти: $k \in \{1, \ldots, n\}$, такое, что данное вхождение слова W (вероятнее всего) соответствует толкованию s_k.

1. $W \leftarrow$ *Лемма* (W)
2. для всех $x \in S \cup H \cup s_1 \cup \ldots \cup s_n$ повторять
3. $x \leftarrow$ *Лемма* (x)
4. для всех $i = 1, \ldots, n$ повторять
5. *близость* $(i) \leftarrow 0$
6. для всех $x \in (S \cup H) \setminus \{W\}$ и всех $y \in s_i$ повторять
7. *близость* $(i) \leftarrow$ *близость* $(i) + w(x,y)$
8. $k \leftarrow$ *argmax* (*близость* (i))
9. Функция $w(x,y)$
10. если $x = y$, то
11. $w \leftarrow 1,0$
12. иначе, если $x \in$ *Синонимы* (y) или $y \in$ *Синонимы* (x), то
13. $w \leftarrow 0,5$
14. иначе, если $x \in$ *Дериваты* (y) или $y \in$ *Дериваты* (x), то
15. $w \leftarrow 0,5$
16. иначе
17. $w \leftarrow 0$
18. если $x \in$ *Статья* (y) или $y \in$ *Статья* (x), то
19. $w \leftarrow w + 0,1$

Как уже было сказано в конце п. 3.2, в случае равных значений близости в строке 8 выбирается наименьший номер k.

4. ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ

Мы применили разработанный алгоритм разрешения неоднозначности значений слов в толковых словарях к толковому словарю испанского языка группы Анайа, содержащему около 30.000 заглавных слов. Среднее количество слов в толковании одного значения составляет 8,39 знаменательных слов (т. е. служебные слова не считаются). Данный словарь является обычным толковым словарем со всеми проблемами, связанными с неточностями и порочными кругами в толкованиях, а не специально подготовленным словарем с ограниченным подмножеством слов в толкованиях, как, например, Longman Dictionary of Contemporary English (LDOCE).

Мы использовали следующие дополнительные лингвистические данные. Для определения того, является ли слово синонимом другого слова, использовался словарь синонимов испанского языка, содержащий около 20.000 заглавных слов. Для определения того, является ли слово морфологическим дериватом, использовалась упрощенная модель словообразования испанского языка. Эта модель решает, являются ли слова дериватами, на основании проверки совпадения по меньшей мере первых пяти символов в словах, например, presidente 'председатель' и presidir 'председательствовать'. В дальнейших исследованиях, разумеется, необходимо использовать более содержательную словообразовательную модель.

В качестве основы для оценки результатов (baseline) были также реализованы и применены к тому же самому словарю два других алгоритма: 1) алгоритм Леска в исходной форме — т. е. без вычисления весов, а именно, $w(x,y) = 1$, если $x = y$, иначе $w(x,y) = 0$; и 2) алгоритм, который всегда выбирает первое значение из списка значений слова соответственно порядку, в котором они приведены в словаре. Для оценки результатов работы алгоритмов было случайным образом выбрано 50 заглавных слов и для них произведена ручная проверка.

4.1. Результаты работы алгоритма

Ручная проверка показала, что в процессе предобработки в 92% неоднозначность части речи либо отсутствовала, либо была разрешена правильно. Большая часть ошибок была связана с неправильным разрешением омонимии существительное—прилагательное, весьма частой в испанском языке. Очевидно, что в этом случае правильное разрешение неоднозначности значений невозможно, поэтому мы не учитываем эти данные при подсчете результатов. Но неправильное разрешение омонимии частей речи в этом случае не влияет существенным образом на сам алгоритм, потому что мы используем морфологическую модель, отрабатывающую эти случаи и, кроме того, обычно толкование существительного не сильно отличается от толкования соответствующего прилагательного.

Результаты работы трех алгоритмов выбора значения — двух базовых и предложенного нами — показаны в следующей таблице. Данные приведены только для семантически неоднозначных слов (с двумя и более значениями) с правильно разрешенной неоднозначностью части речи, слова с одним значением или с неправильно установленной частью речи не учитывались.

Алгоритм	Ошибки, %	Хуже, %
Всегда первое значение	29	123
Исходный алгоритм Леска	17	30
Улучшенный алгоритм Леска	13	0

В третьей колонке таблицы показано, на сколько процентов данный алгоритм делает ошибок больше, чем наш. Как видно из таблицы, наш алгоритм допустил 13% ошибок, что почти на треть лучше, чем исходный алгоритм Леска, допустивший 17% ошибок на тех же данных, и более чем вдвое лучше, чем алгоритм, всегда выбирающий первое значение, который дал 29% ошибок.

Интересно, что алгоритм, всегда выбирающий первое значение, показал сравнительно хороший результат (напомним, что учитывались только слова, имеющие как минимум два значения — следовательно, случайный выбор дал бы как минимум 50% ошибок). Это свидетельствует о том, что первое значение, интуитивно выбираемое лексикографом при составлении словаря, действительно является самым частотным.

4.2. Примеры работы алгоритма

Рассмотрим несколько примеров работы алгоритма. Для испанского слова *abadía* (*аббатство*) в словаре есть следующее толкование:

Abadía = Monasterio, territorio y, en general, bienes que gobierna el abad o la abadesa. 'Аббатство = монастырь, территория и все имущество, которыми управляет аббат или аббатиса.'

Слово *abad* 'аббат' имеет три значения:

1. *Título que recibe el superior de un monasterio o el de algunas colegiatas.*
   'Титул, который получает глава монастыря или некоторых учебных заведений.'
2. *Presidente temporal de un cabildo.*
   'Временный президент капитула.'
3. *En algunas provincias, cura.*
   'В некоторых провинциях, священник.'

Первое значение пересекается с рассматриваемым нами текстом (толкованием слова *abadía*): в обоих содержится слово *monasterio* 'монастырь'. Второе значение не имеет никаких пересечений с рассматриваемым текстом. Третье же значение содержит слово *provincia* 'провинция', которое, в свою очередь, имеет в своем толковании слово *territorio* 'территория', входящее в рассматриваемый текст. Применяя изложенный выше алгоритм вычисления весов, получаем 1,0 для первого значения, 0,0 для второго и 0,1 для третьего. Таким образом, заключаем, что в данном контексте слово *abad* употребляется в своем первом значении, что соответствует действительности.

Заметим, что при подсчете весов мы не учитывали, что слово *abad* 'аббат', входящее в рассматриваемый текст, имеет синоним *superior* 'настоятель', входящий в первое толкование. Действительно, поскольку слово *abad* является как раз самим рассматриваемым словом, значения которого сравниваются, учет его связи с текстом одного из толкований дал бы этому последнему постоянное преимущество, не зависящее от контекста.

Рассмотрим другой пример, в котором вес зависит от морфемного сходства слов. Слово *operación* 'операция' в одном из своих значений имеет такое толкование:

Operación = Negociación con valores bancarios.
'Операция = передача банковских ценностей.'

В этом толковании слово *valor* может иметь одно из указанных в том же словаре одиннадцати значений:

1. *Precio, cualidad de las cosas por la que se paga cierta cantidad.*
   'Цена, количество вещей, за которые платится какое-то количество денег.'
2. *Significado o importancia de algo dicho, escrito, etc.*
   'Значимость, важность чего-либо сказанного, написанного и т. д.'
3. *Cualidad del que no teme el peligro.*
   'Качество не бояться опасности.'
4. *Equivalencia, especialmente en monedas con respecto a las tomadas como patrón.*
   'Эквивалентность, особенно между монетами, с принятым за эталон.'
5. *Grado de utilidad, importancia o buenas cualidades de algo.*
   'Степень полезности, важности или положительных качеств чего-либо.'
6. *Atrevimiento, desvergüenza.*
   'Отчаянность, бесстыдство.'
7. *Firmeza, integridad.*

'Твердость, целостность.'

8. *Eficacia.*
   'Эффективность.'
9. *Duración de una nota musical.*
   'Длительность музыкальной ноты.'
10. *Acciones, bonos o cualesquiera documentos negociables, acreditativos de una propiedad.*
    'Акции, боны или другие ценные бумаги, удостоверяющие владение.'
11. *Persona que posee cualidades positivas para algo determinado.*
    'Тот, кто обладает положительными качествами для чего-либо.'

Единственное значение, имеющее пересечение с рассматриваемым текстом — десятое: оно имеет совпадение negociable 'подлежащий передаче' и negociación 'переговоры, передача', устанавливаемое с использованием модели словообразования. Таким образом, мы заключаем, что, в толковании слова *operación* 'операция' слово *valor* употреблено в десятом значении 'ценные бумаги'. Заметим, что исходный алгоритм Леска не нашел бы указанного совпадения.

Как и в предыдущем примере, синонимия слов *bono* 'бон' и *acción* 'акция', встретившихся в десятом значении, слову *valor* не учитывается, поскольку это бы дало данному значению неоправданное постоянное преимущество.

Разберем подробно еще один пример: слово *abajo* в значении

Abajo = Hacia un lugar o dirección más bajo.
'Вниз = в сторону более низкого места, в направлении низа.'

Рассмотрим выбор значений для слова *dirección* 'направление' в этом определении. Слова *hacia* 'к', *un* — артикль, *o* 'или' и *más* 'более' являются служебными и игнорируются при работе алгоритма. В приведенном ниже примере в начале строки указаны найденные нашим алгоритмом слова. Слова, имеющие точное пересечение с толкованием, указаны без каких-либо дополнительных символов; в "лапках" <> стоят синонимы, в квадратных скобках [ ] — слова, пересекающиеся с толкованием слов, входящих в рассматриваемое толкование. Полученный согласно алгоритму вес указан справа от значения.

dirección:

1. [*lugar*], <*posición*>, <*espacio*>, <*dirección*>, <*línea*>, <*dirección*>: Posición en el espacio de la línea que señala el avance de un cuerpo en movimiento — 2,6
   '[место], <положение>, <пространство>, <направление>, <линия>, <направление>: Положение в пространстве относительно линии, обозначающей направление движения.'
2. [*lugar*]: *Señas escritas en una carta o envío.* — 0,1
   '[место]: Указания на письме или посылке.'
3. [*lugar*]: *Acción y efecto de dirigir o dirigirse.* — 0,1
   '[место]: Действие и результат движения.'
4. *Mecanismo para guiar los automóviles.* — 0,0
   'Механизм управления автомобилем.'
5. [*lugar*], <*dirección*>, <*domicilio*>: *Domicilio de una persona.* — 1,1
   '[место], <направление>, <место жительства>: Место жительства кого-либо.'
6. [*lugar*], <*dirección*>, <*dirección*>: *Cargo y oficina del director.* — 1,1
   '[место], <дирекция>, <дирекция>: Должность и офис директора.'
7. <*dirección*>: *Conjunto de individuos que están al mando de una empresa, organismo, asociación o partido.* — 0,5
   '<дирекция>: Люди, управляющие предприятием, ассоциацией или партией.'
8. *Técnica de realizar una película, en su aspecto artístico o de producción.* — 0,0
   'Производство кинофильма в творческом или техническом аспекте.'

Как видно, для слова *dirección* будет выбрано первое значение, что соответствует действительности. Заметим, что алгоритм Леска без учета синонимов не смог бы выбрать между значениями, потому что прямых пересечений между ними и словами текста (толкования) нет.

5. ВЫВОДЫ

В статье представлен алгоритм разрешения неоднозначности значений слов в толковых словарях и описано его применение к толковому словарю испанского языка, содержащему более 30.000 заглавных слов. Алгоритм основан на идее алгоритма Леска, по сравнению с которым внесены следующие улучшения: при вычислении весов отдельных значений используются 1) синонимия слов, определяемая по достаточно большому словарю синонимов (мы использовали словарь синонимов испанского языка, содержащий более 20.000 заглавных слов); 2) модель словообразования (мы использовали упрощенную модель); 3) толкования слов, входящих в рассматриваемое толкование (в нашей реализации алгоритма эти толкования берутся из того же словаря, к которому применяется алгоритм).

Применение алгоритма разрешения неоднозначности значений слов к толковому словарю позволяет упростить структуру алгоритма, например, исчезает проблема размера окна контекста.

Представленный алгоритм дает лучшие результаты, чем алгоритмы, использованные для сравнения — алгоритм Леска в исходной форме и алгоритм, всегда выбирающий первое по порядку значение слова.

Возможные пути улучшения полученных результатов состоят в проведении экспериментов для более точного вычисления веса каждой части алгоритма (вместо используемых в данный момент эмпирически выбранных значений параметров 1,0, 0,5 и 0,1, см. п. 3.2), в использовании более содержательной модели словообразования и в привлечении дополнительных источников информации.

* * *

Работа выполнена при частичной поддержке правительства Мексики (CONACyT, SNI, CGPI-IPN), правительства Республики Корея (кафедра KIPA для приглашенных преподавателей в Корее) и ITRI Университета Чунг-Анг. Во время подготовки статьи первый автор находился в командировке в Университете Чунг-Анг. Work done under partial support of Mexican Government (CONACyT, SNI, CGPI-IPN), Korean Government (KIPA professorship for visiting faculty positions in Korea), and ITRI of Chung-Ang University. During the preparation of this paper the first author was on Sabbatical leave at Chung-Ang University.

СПИСОК ЛИТЕРАТУРЫ

1. Gelbukh A. F. Using a semantic network for lexical and syntactical disambiguation // CIC-97, Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación, Simposium Internacional de Computación.— Mexico: Mexico City, 1997.— P. 352–366.

2. Manning C. D., Shutze H. Foundations of statistical natural language processing.— Cambridge, MA: The MIT press, 1999.— 680 p.

3. Jurafsky D., James H. Martin Speech and Language Processinf.— N. Y.: Prentice Hall, 2000.— 934 p.

4. Lesk M. Automatic sense disambiguation using machine-readable dictionaries: how to tell a pine cone