УДК 681.3 : 519.76

МЕТОДЫ И АЛГОРИТМЫ ПОСТРОЕНИЯ СЕМАНТИЧЕСКОГО ОБЪЕМА СЛОВА

Ю.Ю. Черепанова

В статье исследованы вопросы построения семантического объема слов с целью дальнейшего его использования для формального описания семантики естественного языка. Разработаны метод и алгоритм построения семантического объема по дефинициям и алгоритм семантического кодирования, являющийся развитием существующего алгоритма «Свертка»

1. Введение

В настоящее время все также актуальной является задача машинного понимания естественного языка, в частности, описания его семантики. Одним из способов описания семантики языка является построение его полевой структуры.

Для построения парадигматического семантического поля наиболее эффективным представляется метод компонентного анализа [1], в процессе которого каждому слову, понятию приписывается набор семантических признаков, представляющих собой комплекс элементарных смыслов, входящих в значение слова. Следующим шагом компонентного анализа является анализ семантических признаков слов с целью проверки каких–либо гипотез об их семантике. В полевой теории такой анализ используется для нахождения семантических связей между словами. В данной статье будет рассмотрен первый шаг компонентного анализа – построение семантического объема слова (набора семантических признаков, составляющих его значение).

Цель работы – разработать метод построения семантического объема с возможностью наиболее полной автоматизации.

2. Методы построения семантического объема слов

В литературе встречаются различные способы выделения семантических признаков в значениях слов. Одним из методов, который называют психолингвистическим, является нахождение пар слов, находящихся в оппозиции друг к другу и выделение компонента значения, образующего эту пару, в качестве одного из семантических множителей. Таким образом формируется «банк семантических компонентов», из которых далее в семантический объем слова выбираются нужные компоненты [2]. Достоинством этого метода является то, что он формирует довольно ограниченный набор семантических компонентов, что является преимуществом в компонентном анализе. Однако, так как таких упорядоченных пар в языке не так много, то данный метод представляется не таким результативным. Раз система языка не такая стройная и упорядоченная, то не все нюансы значения могут быть выделены с помощью метода оппозиции. А раз так, то и

построенные семантические объемы будут не полностью отражать семантику слова. Кроме того, так как автоматический семантический анализ в данное время оставляет желать лучшего, то выделение таких компонент должно проводится вручную, и, следовательно, будет обладать большой субъективностью, и низкой производительностью. Построение семантических полей представляет реальный практический интерес тогда, когда производится большой охват лексики, и снижен процент шума, вносимый субъективным подходом.

Поэтому многие исследователи склоняются к использованию более общего метода – анализу дефиниций слов [3-5]. Правомочность такого подхода объясняется следующим:

1.Словарные дефиниции представляют собой четкие определения, являющиеся плодом коллективного труда ряда разработчиков-экспертов, что гарантирует их объективность;

2.Так как знания человека вербальны (каково бы ни было внутреннее представление знаний, но о них всегда можно рассказать, сохраняются знания и передаются следующим поколениям также в виде написанных текстов), то можно утверждать, что книги, призванные обучать языку, адекватно отражают семантику слова.

При анализе дефиниций для построения семантических объемов слова возникает несколько путей:

– Выделение в качестве семантических множителей слов определения [3];

– Выделение в качестве множителей лексических оборотов - словосочетаний и более крупных конструкций, обладающих более конкретным, узким смыслом [5].

– Различение в определении семантических компонентов, обладающих разными ролями (например семы, обладающие номинативным значением, и релятемы, указывающие, как связываются между собой семы)[5].

Предлагаемый в данной статье метод представляет собой комбинацию первых двух подходов. Третий путь не используется, так как нулевые множители (к которым относятся и местоимения, и предлоги, и союзы) предполагается не включать в семантический объем, а различение роли значащих единиц смысла по сути дела дискредитирует саму идею представления семантики слова как некоторого множества равноправных единиц. Нельзя универсально (правомочно в любых случаях) выделить в качестве вспомогательных (связывающих), например, какие-либо части речи. К примеру, в падежах Филмора [6] в качестве главного выделяется глагол, но нельзя же сказать, что прилагательное является второстепенной частью речи в представлении (выражении) смысла в таких сложных лингвистических системах, как тексты на естественном языке. С другой стороны, в семантических сетях в качестве связей предлагается использовать глаголы. Как мы видим, уже множество различных взглядов на эту проблему доказывает неправомочность подхода деления лексических единиц, по сути принадлежащих к одному уровню, на "главные" и "вспомогательные".

Например, в предложениях: "В комнате тепло", "В комнате теплеет", "Теплая комната", по сути представлена почти одна семантическая информация (в первом и третьем случаях), или, по крайней мере, сходная (в первом и третьем случаях - констатация уже теплой комнаты, во втором – процесс увеличения температуры). Таким образом, нельзя оставить за какой-либо одной частью речи главенствующую роль, уделив остальным лишь роль связки. Также данный пример показывает, что нельзя разделять семантические роли, ориентируясь и на роль слова или оборота в предложении. В настоящее время не известно о существовании стопроцентного синтаксического и семантического анализатора, поэтому нельзя при разделении семантических ролей опираться на синтаксические данные.

Учитывая все выше сказанное, разработаем алгоритм построения семантического объема по дефинициям слов, используя в качестве мельчайших элементов смысла (но не элементарных, так как в большинстве своем эти единицы могут иметь свои дефиниции и, соответственно, семантический объем) семантические множители, соответствующие словам дефиниций (элемент первого подхода). Однако на следующем шаге компонентного анализа для устойчивых словосочетаний, внесенных в словарь терминов и имеющих свое определение, в качестве анализируемой следует взять дефиницию словосочетания, а не определения слов, входящих в него (что частично соответствует второму подходу). В алгоритме будем использовать теоретико-множественную формализацию семантических полей, введенную в работе [7].

3. Алгоритм построения семантического объема слова по его дефиниции

Данный алгоритм построен с учетом того, что слово имеет одно определение. Если слово имеет несколько значений, то есть два пути построения семантического объема слова:

– применение данного алгоритма для каждого значения слова, рассматривая разные значения как разные слова;

– объединение всех определений слова в одну дефиницию, построение обобщенного семантического объема слова, включающего в себя семы всех значений.

На входе алгоритма – слово v_i, семантический объем $W_i = \varnothing$.

1-й шаг Получение определения $v_i \rightarrow o_i$.

2-й шаг Получение множества словоформ, участвующих в тексте определения: Пусть есть такое отношение $R_{раз} \subset O \times S$, что его сечением $R_{раз}(o_i)$ является множество $\{s_{i1},\ldots,s_{iz}\}$, которое представляет собой набор всех словоформ, входящих в текст определения.

3-й шаг Лемматизация (приведение слов в исходную форму), отсечение слов с нулевым значением: Результатом данного шага является множество слов $\{v_{i1},\ldots,v_{il}\}$, являющихся сечением отражения $f_{лем} : S \rightarrow V$ (представляющем

собой приведение слов в исходную форму) по множеству $\{s_{i1},\ldots,s_{iz}\}$ причем $l \le z$.

4-й шаг Выполняется кодирование слов в квазиосновы, результатом которого является множество семантических признаков $\{a_{i1},\ldots,a_{ik}\}$, являющихся сечением $f_{код}(v_{i1},\ldots,v_{il})$, причем $k \le l$.

5-й шаг Добавление полученных семантических признаков в семантический объем: $W_i = W_i \bigcup \{a_{i1},\ldots,a_{ik}\}$.

6-й шаг Если уровень анализа достаточен, то выход. Если следующий уровень анализа необходим, то получить множество слов $\{v_{i1},\ldots,v_{ik}\}$, являющихся сечением отражения $f_{дек}: A \to V$ (представляющем собой приписывание каждому признаку слова, наиболее типично представляющего его смысл) по множеству $\{a_{i1},\ldots,a_{ik}\}$, и выполнить шаги 1-6 по каждому слову v_{ij}.

На выходе - множество $W_i = \{a_{i1},\ldots,a_{ik}\}$, представляющее собой семантический объем слова v_i.

Поясним некоторые шаги данного алгоритма.

Необходимость операции декодирования, которая применяется на шестом шаге, объясняется тем, что при кодировании слов в квазиосновы, одна квазиоснова может представлять несколько слов, в соответствии с требованиями уменьшения словаря множителей для более эффективного представления смысла (суть требования – чтобы слова, семантически означающие почти одно и то же должны кодироваться в один семантический множитель). Однако при увеличении глубины анализа необходимо связать множитель с дефиницией. Наиболее предпочтительным (и легко реализуемым) способом может быть метод связывания с семантическим множителем слова, содержащего усредненное определение из всех определений слов, которые кодируются в данный множитель. Однако данный метод имеет свои недостатки. Во-первых, не всегда может существовать такое слово, следовательно, он будет достаточно приблизительным. Кроме того, приписывание такого слова может быть не всегда правомерным. Другим методом, более точным и экономящим время, может быть связывание семантического множителя сразу с набором других множителей, входящих в усредненное множество среди семантических объемов первого уровня слов, кодируемых в данный множитель.

4 Анализ существующих методов кодирования слов в семантические множители

Существуют разные пути кодирования семантической информации в дефиниции, и каждый из этих путей обладает известными преимуществами, но имеет и какие-то недостатки. Первым, самым простым решением, казалось бы, было выделение в качестве кода корня слова. Однако словообразовательные отношения настолько сложны, и приставочные модификации от одного и того

же корня могут в такой степени изменить значение слова, что семантическая связь нарушится (*сказать – присказка*, *гадать – загадка* и т.п.).

Вторым путем кодирования является лемматизация, т.е сведение словоизменительных форм к исходной форме: существительного – к именительному падежу единственного числа, прилагательного – к форме именительного падежа единственного числа мужского рода, глагола – к инфинитиву и т.п. Хотя лемматизация может осуществляться машинным способом, такой путь представляется малоэффективным, поскольку приводит к чрезмерному росту числа семантических множителей, и кроме того, в этом случае семантически явно связанные между собой единицы (например, «*болезнь*» и «*болезненный*») выступают как разные, самостоятельные и отличающиеся друг от друга, а они могут даже не иметь разных дефиниций [3]. Кроме того, применение данного метода приводит к большому росту мощности машинных словарей, применяемых для лемматизации.

Третьим методом, довольно успешно применявшимся в работе [3], было построение квазиоснов. Был принят такой принцип, что начало слова оставалось без изменения, а сокращению, усечению подвергалась правая часть (например, *сказать* – «*сказ*», *присказка* – «*присказ*», *гадать* – «*гада*», *загадка* – «*загад*»). «Такой прием кодирования позволяет не считать релевантным различия между частями речи, образованными от одного корня, и некоторые другие словообразовательные отношения» [3, стр. 42]: «*гада*» - *гадание*, *гадать*, *гадалка*, «*загад*» - *загадка*, *загадывать*. «Правая граница кода никак не связана с морфемным членением слова, она может проходить и внутри корня («*поздр*» – *поздравить*, *поздравление*), и по его конечной букве («*лад*» - *ладно*, *ладить*, *ладный*), и справа от нее («*представи*» – *представитель*(*ство*), *представитель(ный*))» [3, стр. 45]. Таким образом, получаемый сегмент не является ни корнем слова, ни его основой. «Сегмент – это своего рода квазиоснова, исключительно семантический идентификатор слова, и способ его получения именно таков, что он выполняет одну-единственную функцию – быть единицей смысла, однозначным средством для опознавания семантики слова» [3, стр. 46]. Поэтому можно утверждать, что план выражения квазиосновы (побуквенный состав) равен, тождественен его содержанию. Иными словами, сегмент обладает свойством одноплановости, каким и должен обладать семантический множитель. Данный метод уже был применен на практике, однако автор не указывает степень его автоматизации. А между тем, не совсем понятен алгоритм отсечения правой части слова. Ведь если применялись машинные словари, то это довольно громоздкий механизм (как в плане машинного обеспечения, так и в плане времени обработки). И, вообще, в целом, данный алгоритм не выглядит достаточно прозрачным, загроможден, и его универсальность не очевидна.

Четвертый путь, также находящий широкое применение при автоматической обработке текстов, основан на статистической закономерности, в соответствии с которой наибольшей информативностью в русском слове обладают согласные. Этот путь предполагает свертывание сравниваемых единиц, получение так называемых сверток из одних согласных, при этом

гласные первого слога, как правило, сохраняются (например, «*болезнь*» - «*болзн*», «*болезни*» - «*болзн*», «*болезнями*» - «*болзнм*»). Но для решения данной задачи этот способ тоже не очень подходит, так как помимо недостатков, общих с предыдущим, он чреват появлением значительного числа омонимичных кодов – сверток («*стол*» – «*стл*», «*стул*» – «*стл*» и т.п.), а в некоторых вариантах алгоритма свертывание приводит и к значительной синонимии сверток, когда словоформы одной лексемы преобразуются в разные свертки («*болезни*» - «*болзн*», «*болезнями*» - «*болзнм*»)[3].

Возьмем за основу метод четвертого типа, вариант алгоритма которого описан в [8]. Данный алгоритм обеспечивает эффективную свертку существительных и прилагательных с довольно низким уровнем синонимии и омонимии сверток, прозрачен и легок в применении. К основным его недостаткам можно отнести следующее:

– Не рассматривает глаголы, причастия, и т.д., хотя различные части речи обладают одинаковыми семантическими качествами.

– Не всегда исключает синонимию сверток (*конь –конь, конем-кон*).

– Сохраняет недостаток, присущий четвертому подходу, не исключая омонимию сверток.

Таким образом, необходимо разработать алгоритм кодирования, основанный на вышеуказанном алгоритме «Основа», в который следует внести следующие изменения:

– Исправить список отсекаемых окончаний (добавить "й", "ь").

– Разработать механизм свертки различных форм глагола.

– Разработать механизм разрешения неоднозначности (омонимии) кодов.

– Разработать механизм разрешения синонимии кодов.

Очевидно, что последние два требования необходимы уже при использовании полученных семантических множителей (например, для процесса декодирования, увеличении глубины семантического анализа, при сравнении полученных семантических объемов с целью установления соответствия).

5. Алгоритм кодирования слов в семантические множители.

Предлагаем следующий алгоритм кодирования слов в семантические множители. На входе – кодируемое слово. На выходе – свертка, семантический множитель.

1-й шаг Две первые буквы слова включаем в свертку;

2-й шаг Начиная с третьей буквы, включаем согласные слова в свертку (общая длина свертки - до 6 букв);

3-й шаг Убираем конечные согласные свертки "в", "м", "х", "ь", "й" (и их скопления в конце свертки);

4-й шаг Отсекаем у свертки окончания глаголов в соответствии с таблицей 1 (если в кодируемом слове на расстоянии от конца слова, указанном в четвертой колонке, после гласной есть сочетания, указанные в первой колонке, то убираем из свертки сочетание из третьей колонки).

Таблица 1 Отсекаемые окончания при свертке глагольных форм

Сочетания в конце слова	Примеры окончаний	Отсекаемые части свертки	Расстояние от конца слова
ться	-аться, -иться, -еться	тьс	0
тся	-ется, -ятся	тс	0
ть	-ать, -ять, -еть, -ить	т	0
т	-ет, -ют, -ят, -ит	т	0
шь	-ешь, ишь	ш	0
чь	-ечь	ч	0
л	-ел –ал	л	0
щего	-щего	щ	0
щ	-щее, -щую, -щие	щ	2
щ	-щиеся	щ	4
щегося	-щегося	щгс	0
го	-ого, -его	г	0

5-й шаг Если длина получившейся свертки менее трех букв, то свертка равна трем первым буквам слова.

6-й шаг Если длина свертки более 6, то оставляем в свертке первые 6 букв, остальные отсекаем.

Поясним модифицированные шаги разработанного алгоритма. Введенные в список отсекаемых на третьем шаге согласные позволяют устранить окончания прилагательных и существительных (например, было: *быстрый-быстрй, быстрая – быстр*, стало *быстр* в обоих случаях).

Введенный в таблице список окончаний позволяет избежать синонимии кодов глагола (например, *отвечать, отвечал, отвечаешь* кодируются в один код *отвч*), и преобразовать части речи, скорее всего не имеющие собственных дефиниций, к родственным, семантически близким частям речи, имеющим свои дефиниции (например, *смотрящий* кодируется в код *смтр*, как и глагол *смотреть*). Данный метод позволяет отсекать окончания, не вводя (и не сравнивая) весь список окончаний (например: "-ал", "-ала", "-ел", "-ет", "-ют", "-ят"), группируя их на семейства, что уменьшает затраты памяти и, что важнее, времени на обработку (уменьшая трудоемкость по крайней мере в 3 раза). Описанное правило позволяет отличать ситуации, когда рассматриваемое сочетание встречается в составе суффиксов и окончаний глаголов от их вхождений в другие части речи, например, существительные (вхождение сочетания "-ть" в окончание существительных, обозначающих свойство: *организованность, подчиненность, принадлежность,* во флективное окончание существительного третьего рода: *сутью*), в которых данные сочетание могут иметь семантическую нагруженность, в то время как в глаголах представляют собой часть флексии, отсечение которой необходимо для уменьшения синонимии кодов.

На этом же шаге отсекаются окончания *–ого, -его*. В алгоритме «Основа» приведенном в [8], буква "г" включена в состав букв, отсекаемых на 3-м шаге. Однако проведенные испытания показали, что отсечение конечных букв "г" увеличивает процент синонимичности кодов, в то же время для правильного кодирования прилагательных следует учесть отсечение окончаний *"–ого", "-его"*. Разработанный механизм отсечения окончаний глагольных форм позволил также правильно распознавать и эти окончания прилагательных.

Однако данный метод не позволяет полностью устранить омонимичность сверток. Но наличие разработанных механизмов разрешения неоднозначности, описанных в работе [9], позволяет считать достигнутый уровень «антиомонимичности» кодов приемлемым.

Достоинством данного метода семантического кодирования является то, что он позволяет в некоторой степени уменьшить количество ошибок. В частности, не являются важными при использовании такого метода кодирования ошибки (описки) в согласных и гласных отсекаемых окончаниях, в гласных основы.

7. Выводы

В статье исследованы вопросы построения семантического объема слова. Рассмотрены недостатки существующих методов, разработаны метод и алгоритм построения семантического объема по дефинициям слов, заключающийся в итерационном разбиении текста на элементы смысла.

Проанализированы методы семантического кодирования для получения семантических множителей и разработан алгоритм семантического кодирования, являющийся развитием существующего алгоритма «Свертка».

Разработанные методы и алгоритмы могут быть применены для описания семантики слов естестественного языка, например при построении концептуальных моделей ответов на естественном языке при тестировании знаний [10]. Также данные методы могут быть эффективно использованы при поиске информации.

Литература

1. *Павлов П. Ф., Черепанова Ю. Ю. Шубин И. Ю.* О возможности построения тезауруса семантических полей и его применения в информационных системах // Вісник ХДПУ. Збірка наукових праць. – Вип. 108. – Харьков: ХДПУ, 2000. – С. 41-46.
2. *Лесохин М М., Лукьяненков К. Ф., Пиотровский Р. Г.* Введение в математическую лингвистику: Лингвистическое приложение основ математики. – Мн.: Наука и техника, 1982. – 263 с.
3. *Караулов Ю. Н.* Частотный словарь семантических множителей русского языка. – М.: Наука, 1980. – 207 с.
4. *Арнольд И. В.* Основы научных исследований в лингвистике: Учебное пособие. – М.: Высшая школа, 1991. – 140 с.
5. *Скороходько Э. Ф.* Семантические сети и автоматическая обработка текста. – К.: Наукова думка, 1983. – 284 с.

6. *Гладкий А. В., Мельчук И. А.* Элементы математической лингвистики. – М.: Наука, 1969. – 192 с.
7. *Черепанова Ю. Ю.* О теоретико-множественном и теоретико-категорном подходах к моделированию семантических полей // Проблемы бионики: Всеукр. межвед. науч.-техн. сб. – 2001. – Вып 54. – С.75-78.
8. *Захаров В. П., Мордовченко П. Г., Сахарный Л. В.* Совершенствование лингвистического обеспечения в ИПС «бестезаурусного» типа // НТИ. – Сер. 2. – 1980. – №6. – С.14-17.
9. *Черепанова Ю. Ю.* Применение компонентного анализа для разрешения неоднозначности естественного языка // Труды 6-го Междунар. Молодежн. форума "Радиоэлектроника и молодежь в XXI веке": Сб. научн. трудов. – Ч.2. – Харьков: ХНУРЭ. – 2002. С. 484-485.
10. *Черепанова Ю.Ю.* Тестування теоретичних знань за допомогою тезауруса семантичних полів. // Вісник Національного університету “Львівська політехніка” “Інформаційні системи та мережі”. – 2002. – №464. – С. 318-326.