

Э.Н. МАНУКЯН, А.С. МАНУКЯН

ОБ ОРГАНИЗАЦИИ СЕМАНТИЧЕСКОГО АНАЛИЗА ПРИ МАШИННОМ
ПЕРЕВОДЕ

Предлагаются эффективные алгоритмы семантического анализа, использующие специфические особенности базы знаний системы машинного перевода и обеспечивающие высокую скорость реализации решения.

Ключевые слова: алгоритмы семантического анализа, машинный перевод, база знаний.

При машинном переводе (МП) после завершения этапа морфологического анализа слова входного предложения заменяются концептами базы знаний (БЗ) системы – смысловыми аналогами слов [1]. При последующем синтаксическом анализе организуется генерация набора возможных синтаксических отношений выбранного претендента–концепта с соседними концептами входного предложения. При наличии омонимии [2] каждому слову ставится в соответствие множество концептов, в связи с чем синтаксический анализ существенно осложняется ввиду необходимости выбора наиболее вероятного претендента из указанного множества. Наилучшим считается претендент, имеющий наиболее тесные синтаксические и семантические отношения с соседствующими с ним словами предложения.

Для выбора наиболее правдоподобного элемента созданного набора организуется подэтап семантического анализа, при котором для каждого из сгенерированных отношений вычисляется некая количественная мера, называемая коэффициентом правдоподобия. Учитывая требования, предъявляемые к эксплуатационным характеристикам систем МП и к качеству перевода, физическая и логическая организации БЗ системы должны быть спроектированы максимально приспособленными (обеспечивающими наименьшие временные потери) для вычисления оценок коэффициентов правдоподобия.

В системах МП, как правило, используется сетевая логическая организация базы, при которой:

- узлы сети представляют концепты предметной области;
- атрибутами служат имена концептов на переводимых языках;
- ветви соответствуют системным отношениям типа вид-класс, система-подсистема, причина-следствие и т.д. Они могут выступать в качестве семантических аналогов некоторых основных синтаксических отношений, например, таких как: отношение подлежащее – сказуемое, прямые и косвенные дополнения и т.п.

Коэффициенты правдоподобия могут быть оценены в ходе проверки наличия искомых синтаксических отношений между двумя заданными концептами.

К сожалению, учет всевозможных синтаксических отношений для всех концептов в БЗ переводчика делает его громоздким и практически неприемлемым для использования в реальном масштабе времени. По этой причине проектировщиками БЗ в качестве допустимых отношений при построении сети выбирается только часть синтаксических отношений, а проверка наличия нефиксированных в базе отношений между концептами осуществляется путем поиска путей, проходящих через другие концепты БЗ. Составляются эти траектории движения из имеющихся в базе бинарных отношений.

При навигациях в сети БЗ как на типы отношений, так и на возможности их соседства в сети ставятся определенные ограничения. В частности, проблемы возникают при организации проверок наличия отношений между существительными и определяющими их притяжательными или относительными прилагательными, поскольку установление непосредственных отношений между концептами, представляющими эти части речи, невозможно ввиду значительного их количества. Решение этой и подобных задач требует определения наикратчайшего пути [2,3] между двумя узлами сети с заданными весовыми коэффициентами дуг и коэффициента правдоподобия искомого синтаксического отношения.

Отметим, что применение предлагаемых до настоящего времени подходов для систем МП неприемлемо, что обусловлено громоздкостью семантической сети, включающей десятки тысяч концептов со своими связями, и высокими требованиями, предъявляемыми к быстродействию системы. В связи с этим предлагается метод решения задачи, эффективно использующий специфические особенности БЗ системы МП и обеспечивающий требуемую скорость реализации решения.

Суть метода заключается в следующем.

Каждому i -му узлу семантической сети предлагается приписать обсуждаемый ниже список констант C_i . Если требуется найти наикратчайший путь между узлами s и g , то организуется многошаговый процесс поиска этого пути, начиная с узла s . На каждом текущем шаге поиска пути предлагается определить наилучшее его направление, исходя из условия

$$\min_j D(C_j, C_g), \quad (1)$$

где j – номер узла, имеющего непосредственное отношение к узлу сети, находящегося в конце уже определенного фрагмента пути; $D(C_j, C_g)$ – функция, определяемая методом, обеспечивающим пропорциональность значения $D(.)$ от числа дуг, лежащих на ожидаемом кратчайшем пути между j -ым и g -ым узлами сети.

Рассмотрим задачу определения функции $D(.)$ и опишем правила составления списка C_i , при которых выражение (1) позволит синтезировать наикратчайший путь от s до g .

Для достижения цели предлагается каждому i -му узлу сети БЗ ставить в соответствие некоторую точку в n -мерном пространстве, а в качестве C_i использовать список координат этой точки: $x_1^i, x_2^i, \dots, x_n^i$. Предположим, правило расстановки обеспечивает условие

$$\text{если } R_{ij} > R_{ik}, \text{ то } L_{ij} > L_{ik}, \quad (2)$$

где R_{ij} – расстояние между точками пространства, где расположены i -й и j -й узлы; L_{ij} – количество дуг наикратчайшего пути между j -ым и i -ым узлами сети.

Тогда в качестве $D(.)$ можно принять функцию определения “геометрического” расстояния между двумя узлами n -мерного пространства:

$$D(C_i, C_j) = R_{ij} = \sqrt{\sum_{k=1}^n (x_k^i - x_k^j)^2}. \quad (3)$$

При программной реализации данного подхода модифицируется алгоритм расстановки узлов сети в пространстве и порядок вычисления $D(.)$. При расстановке точек в пространстве предлагается составить n -мерную пространственную решетку, применяя только точки с целочисленными значениями координат, а в качестве $D(.)$ использовать функцию

$$D(C_i, C_j) = \sum_{k=1}^n |x_k^i - x_k^j|. \quad (4)$$

Это выражение не противоречит условию (2) и по сравнению с (3) легче вычисляемо. Принятые допущения позволяют констатировать, что $D(C_i, C_j) = \mu$, если $L_{ij} = 1$, где μ - шаг решетки.

Таким образом, задача определения $C_i (i=1, 2, \dots)$ сводится к задаче определения размерности n и правила расстановки вершин сети по точкам решетки так, чтобы удовлетворялось условие (2) для всех узлов сети.

Предлагается подход, согласно которому при расстановке узлов сети в пространстве будет сформирована некая замкнутая подобласть, отвечающая заданным условиям. Из граничных узлов заданной подобласти выбирается узел, имеющий максимальное количество соседей, находящихся вне замкнутой области. Эта замкнутая подобласть наращивается. Формируются ее новые границы путем специальной перестановки в пространстве указанных соседей. Максимальной эффективности процесса расстановки можно достигнуть, если в процессе наращивания подобласти включать в него в первую очередь наиболее связанные узлы сети. Предлагаемый ниже алгоритм реализует такую расстановку.

Алгоритм – 1

1. Определяются максимально связанный узел N сети, искомая размерность пространства $n = FL(\ln/2)$, где $FL(.)$ - функция округления до большего целого; $\ln$ - число дуг, исходящих из узла N . Формируются пустые множества E и V . Узлы, находящиеся в подобласти, включающей уже расставленные узлы, будут представлены множеством E . В подмножество V включаются узлы, находящиеся на границе этой подобласти, и узел N , установленный в начале координат n -мерного пространства.

2. В E добавляется выделенный из V i -й элемент V_i , удовлетворяющий условию $\max_i |G_{V_i}|$, где $G_{V_i} = \frac{G'_{V_i}}{(G'_{V_i} \cap E \cup G'_{V_i} \cap V)}$, а элементы g'_{V_i} множества G'_{V_i} удовлетворяют условию $L_{V_i g'_i} = 1$.

3. Поочередно для каждого элемента g , принадлежащего G_V , определяется точка решетки с координатами $x_r (r = \overline{1, n})$, удовлетворяющая условиям

$$\begin{cases} \sum_{r=1}^n |x_r - x_r^v| = \mu, & \text{если } L_{g,v}(v \in V), \\ \max_{x_r} \left(\sum_{v \in E} \sum_{r=1}^n |x_r - x_r^v| \right). \end{cases} \quad (5)$$

Фактически ищется точка в решетке пространства, для которой ее расстояние от уже расставленных точек не превышает величины шага μ и которая максимально удалена от них. После выявления такой точки ее координаты приписываются узлу $g_i(Cg_i = \{x_1, x_2, \dots, x_n\})$ и элемент g_i включается в множество V .

4. Проверяется состав множества V . Если он оказывается непустым, то выполняется переход к пункту 2, в противном случае - расстановка заканчивается.

Отметим, что реализация 2-го шага алгоритма расстановки не всегда реализуема ввиду возможного отсутствия узла решетки, удовлетворяющего условию (5).

Удачный исход работы данного алгоритма расстановки позволит организовать поиск наикратчайшего пути между узлом N и некоторым узлом M , который явится конкретной реализацией схемы поиска (2). На каждом шагу поиска из множества узлов $P\{P_1, P_2, \dots\}$, соседствующих с данным узлом, выбирается узел P_i , удовлетворяющий условию $\min_{P_i} D(C_N, C_{P_i})$. Поскольку величина $D(\cdot)$ пропорциональна количеству ветвей, находящихся на кратчайшем пути между узлами M и N , то поиск получается направленным и без перебора.

В случае нарушения условий, приведенных на втором шаге данного алгоритма, необходимо провести реорганизацию расстановки узлов сети. Для этого выбирается или путь повторной расстановки после увеличения размерности пространства на единицу, или деформация расставленной сети в пространстве.

Рассмотрим один из возможных алгоритмов деформации.

Пусть на 4-ом шаге алгоритма наблюдаются узлы $m \in G/v_i$ и $n \in V$, для которых $L_{m,n}=1$. Если в решетке отсутствует узел, удовлетворяющий условию (5), то предлагается, не меняя размерности пространства, узел m "стянуть" к узлу n . Поскольку узел m уже имеет определенные связи с предварительно расставленными узлами, то такое "стягивание" приведет к естественному изменению расположения определенных узлов множеств V и E , т.е. к деформации уже ранее расставленного фрагмента сети. Такую деформацию предлагается осуществить по второму алгоритму.

Алгоритм – 2

1. Формируется множество $Q\{n\}$ и пустое множество Q' .
2. Формируется множество C с элементами, удовлетворяющими условию $x_{C_i}^m \neq x_{C_i}^n$.
3. Из Q выделяется элемент q_i и формируется $A\{a_i\}$, где a_i удовлетворяет условиям $L_{a_i q_i}=1$,

(6)

$$\frac{x_i^m}{2} + \frac{x_i^n}{2} \leq x_i^{a_j} \quad \text{для } I \in C,$$

$$x_i^m = x_i^{a_j} \quad \text{для } I \notin C. \quad (7)$$

Отметим, что если найдется хотя бы один узел, не удовлетворяющий условию (7), то деформация невозможна, и алгоритм прерывается.

4. Множество A включается в Q и Q' , а q_i исключается из Q и осуществляется переход к 3-му шагу данного алгоритма, если $Q \neq \emptyset$.

В результате в Q' накапливаются узлы, подлежащие перемещению при деформации сети - "стягиванию" узлов n к узлам m . Значения новых координат определяются по выражениям

$$x_i^{q_i} = \begin{cases} x_i^m + x_i^n - x_i^{q_i} & \text{для } I \in C, \\ x_i^{q_i} \pm 1 & \text{для } I \notin C. \end{cases}$$

Нетрудно убедиться, что после деформации расставленной сети $D(C_m, C_n) \leq \mu$, и она допустима при $2|C| < n$, где n - размерность пространства. Осуществление такой деформации приведет к нарушению условия (2). При этом описанный процесс поиска наикратчайшего пути, где на каждом шаге поиска однозначно определяется оптимальная "навигация" целевого узла, станет неприемлемым. Поэтому приведенный выше алгоритм поиска наикратчайшего пути между узлами M и N можно модифицировать.

Алгоритм – 3

1. Формируются пустые множества F , H и множество P с элементами P_i , удовлетворяющие условию $L_{P_i} = 1$.

2. В множество F включаются элементы $f_i = D(C_{P_i}, C_N)$.

3. В множество H включаются элементы $h_i = D(C_{P_i}, C_M)$.

4. Элементы множеств P , F , H просматриваются с целью удаления из их состава элементов, не удовлетворяющих условию

$$\min_i (f_i + h_i). \quad (8)$$

5. В множество P включаются все j -е узлы сети, удовлетворяющие условию $L_{P_j} = 1$, а соответствующие им значения $D(C_j, C_M)$ и $D(C_j, C_N)$ включаются соответственно в множества H и F .

6. Переходом к пункту 4 данного алгоритма процесс повторяется до достижения узла M .

Заметим, что при идеальной расстановке $D(C_M, C_N) = \mu_{LMN}$. Однако в результате деформации сети, производимой при поиске наикратчайшего пути, значение $D(C_M, C_N)$ уменьшается на некоторую величину Δ . Последнее приводит к появлению ложных путей, что соответствует выполнению условия (1) более чем для одного узла. В результате наблюдается неравенство $|P| \geq 1$ после четвертого шага алгоритма.

С целью оценки количества вершин, лежащих на ложных путях, и принимая во внимание, что все ложные пути, отвечающие условию (7), ограничиваются некой n - мерной призмой с объемом $\Delta^{n-1} D(C_M, C_N)$, введем в рассмотрение понятия: T - количество дуг, лежащих на кратчайшем пути

R- общее количество узлов семантической сети. Тогда оценка количества искомым узлов определится как

$$W = \Delta_{CP}^{n-1} (T - \Delta_{CP}) R / \prod_{i=1}^n X_i \max .$$

Используя выражение для W на 4-ом шаге алгоритма расстановки, можно оценить выборку одного из двух возможных вариантов решений. В первом случае увеличиваются расходы памяти на хранение векторов C, а во втором - временные затраты из-за возникновения ложных путей при «навигации» к целевому узлу.

Полученное выражение позволит дополнительно оценить эффективность предложенного метода для оценки количества искомым узлов W. Например, учитывая, что для широко распространенных «волновых» алгоритмов количество узлов на ложных путях вычисляется выражением: $W' = T^n R / \prod_{i=1}^n X_i \max$, эффективность предложенного метода (по сравнению с «волновым» алгоритмом) можно оценить величиной γ :

$$\gamma = 1 - \frac{W}{W'} = 1 - \left(\frac{\Delta}{T} \right)^{n-1} \left(1 - \frac{\Delta}{T} \right).$$

Существенная эффективность данного алгоритма очевидна. В частности, экспериментально подтверждено, что в пределах $\Delta/T < 0,7$ (что вполне реально при разреженных семантических сетях БЗ) применение предлагаемого подхода, несомненно, становится более целесообразным.

СПИСОК ЛИТЕРАТУРЫ

1. **Потапова К.К.** Новые информационные технологии и лингвистика. – М.: Изд.3. Информация, 2002. – 368 с.
 2. **Вагон В.Г., Кикнадзе В.Г.** Дедуктивный вывод на семантических сетях в системах принятия решения // Изв. АН СССР. Серия тех. кибернетики. -1984. -С. 41-49.
Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение.- СПб: БХВ-Петербург, 2003. – 1104 с.
 3. **Douglas Arnold, Lorna Balkan, Siety Meijer.** MACHINE TRANSLATION. An Introductory Guide, Linguistics, University of Essex, Colchester, CO4 3SQ, UK, doug@essex.ac.uk. -March 6, 2001. – 124 p.
- ГИУА. Материал поступил в редакцию 17.04.2005.

Է.Ն. ՄԱՆՈՒԿՅԱՆ, Ա.Ս. ՄԱՆՈՒԿՅԱՆ ԻՄԱՍՏԱՒԻՑ ՎԵՐԼՈՒԾՈՒԹՅԱՆ ԿԱԶՄԱԿԵՐՊՈՒՄԸ ՄԵՔԵՆԱՅԱԿԱՆ ԹԱՐԳՄԱՆՈՒԹՅԱՆ ՀԱՄԱԿԱՐԳՈՒՄ

Առաջարկվում են սեմանտիկական վերլուծության արդյունավետ ալգորիթմներ, որոնք օգտագործում են մեքենայական թարգմանության համակարգի առանձնահատկությունները և ապահովում որոշում կայացնելու քարձր արագություն:

E.N. MANUKYAN, A.S. MANUKYAN
ORGANIZATION OF SEMANTIC ANALYSIS IN MACHINE TRANSLATION SYSTEM
Effective algorithms of semantic analysis using specific peculiarities of knowledge base in the machine translation system and providing with high speed of solution realization are proposed.