Дымо А.Б.

Взгляд на систему автоматизированного реферирования с точки зрения програмной реализации

Несмотря на существенные проблемы реферирования с лингвистической точки зрения, проблемы програмной реализации на первый взгляд носят характер акциденций, т.е. сопутствующих трудностей. Такое рассмотрение аспектов програмной реализации, несомненно, является верным, так как вышеозначенные лингвистические задачи являются решаемыми (вычислимыми). Однако, как будет показано далее удачное технологическое обеспечение процесса реферирования может повлиять и на некоторые его (процесса) существенные характеристики.

Для начала рассмотрим наиболее очевидные аспекты создаваемой системы. Как было показано ранее, задача реферирования разбивается на задачи синтаксического анализа предложений, структурного анализа предложений и абзацев и компоновки.

Синтаксический анализ может быть выполнен с помощью контекстно-свободного анализатора Link [3], идея которого восходит к модели непосредственно составляющих Бархударова [2, 20]. Независимость от контекста не представляется проблемой для задачи автоматизированного реферирования, так как, во-первых, грамматический центр предложений (тройка субъект-предикат-объект) существует так или иначе вне зависимости от контекста, а, во-вторых, поступающие на вход системы реферирования предложения могут считаться верными с точки зрения семантики, прагматики и связей реального мира. В то же время, нельзя не считаться со значительно большими показателями производительности контекстно-свободных грамматик по сравнению с контекстно-зависимыми. Так, согласно данным [2, 10] производительность алгоритма Link – $O(N^3)$, где N – количество слов в предложении.

Структурный анализ есть:

1) сопоставление частей поступающих предложений шаблонам, соответствующим определенным структурам;
2) сопоставление целых предложений и абзацев правилам согласованности.

Рассмотрение первой из задач структурного анализа приводит к выводу о том, что каждая группа шаблонов для некоторого элемента структуры E_i есть ни что иное, как множество регулярных выражений [1, 101-143], определяющих некий регулярный язык из предложений, соответствующих этому элементу E_i. Програмная реализация алгоритма сопоставления представляет собой недетерменированный конечный автомат (НКА) с ε-переходами с максимальным временем работы $O(N^2)$. Число состояний НКА M равно $O(Rl)$, где Rl – суммарная длина регулярного выражения R, полученного дизъюнкцией выражений всех шаблонов всей структуры E. Ввиду того, что шаблоны могут быть модифицированы пользователем, регулярное выражение R должно преобразовываться в НКА после каждой модификации, можно утверждать, что время сопоставления в процессе структурного анализа не превысит $O(N^2*Rl)$.

Правила согласованности по структуре напоминают грамматику Link, что позволяет применить аналогичную технику анализа (исключая, правда, этап лексического анализа), на этот раз на уровне предложений и целых абзацев. Очевидно, что анализ производится за время $O(Sc^3*Pc^3)$, где Sc – количество предложений в тексте, а Pc – количество абзацев в тексте.

Компоновка предполагает сравнение грамматических центров предложений, соответствующих одному элементу структуры E_i. Само сравнение занимает линейное время. Также линейное время занимает нахождение деноминатора. Т.о. общее время компоновки займет $O(SPc)$, где SPc – количество пар предложений, подлежащих компоновке.

Как показано выше, проблема реферирования является для автоматизированной системы P-проблемой, т.е. разрешимой за полиномиальное время (в нашем случае за $O(Pc*Sc^2*N^5*Rl*SPc))$. Такая характеристика системы, безусловно, является важной. Однако, большое количество определяющих параметров и немалые показатели степени снижают потенциальные области применения. Улучшение же производительности позволит применять предлагаемый метод реферирования в онлайновых поисковых системах, до сих пор использующих менее адекватные, но более производительные статистические методы со временем работы $O(Sc*N*SPc)$.

Представляется возможным подойти к проблеме увеличения производительности системы реферирования с трех сторон, соответствующих синтаксическому анализу, структурному анализу и компоновке. Однако, наиболее очевидный способ лежит в сведении времени сопоставления предложения элементу структуры от $O(N^2*Rl)$ к $O(N)$. Характеристика $O(N)$ в данном случае является целью, т.к. она является скоростью работы человека на той-же задаче. Если задаться целью определить алгоритм сопоставления, в какой-то мере соответствующий процессу, происходящему в человеческом мозге, то можно увидеть путь, ведущий к такой цели. Наиболее удачными средствами моделирования работы мозга есть нейронные сети, описанные, например, в [4].

Нейронная сеть есть весьма удачная аналогия человеческому мозгу применительно к задаче сопоставления предложений элементам структуры по двум причинам. Первая причина – это то, что как и мозг, сеть в состоянии выдать состояния активации (сигналы) в выходном слое нейронов за время $O(N)$ в случае последовательной подачи слов из предложений во входной слой и даже за время $O(1)$ при подаче всего предложения целиком. Второй причиной удачности аналогии есть свойство сетей обучаться и аппроксимировать. Простая сеть с прямой подачей (без реккурентности и самообучения) будучи обучена с помощью разработанной системы автоматизированного реферирования на некотором обучающем множестве предложений L, может затем выдавать заключения о предложениях S_i не из множества L основываясь на “похожести” S_i предложениям из множества L, т.е. выполнять ту же эвристическую процедуру, что и человек в процессе реферирования.

Рассматривая современные топологии нейронных сетей, можно выделить два их вида, соответствующие двум возможным решениям задачи сопоставления.

Первое решение, оно же и наиболее очевидное, состоит в том, что для каждого элемента структуры E_i создается и обучается сеть Хопфильда [5], представленная на рис. 1.

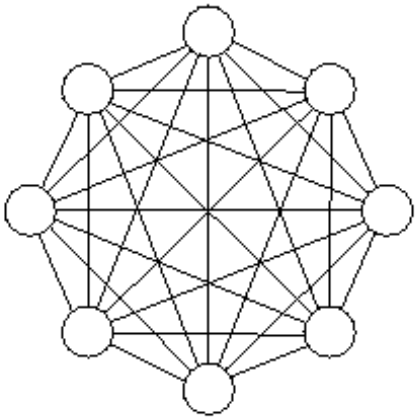

Рис. 1. Нейронная сеть Хопфильда

Т.к. в сети каждый нейрон является одновременно входным и выходным элементом, то слова предложения подаются на вход всех нейронов одновременно, а после обучения каждый нейрон в сети должен выдавать сигнал “1” для всех предложений обучающего множества *L*. В процессе работы такая сеть будет выдавать на всех нейронах сигнал либо “1” либо “0” в зависимости от степени соответствия подаваемых предложений тому элементу структуры, для которого эта сеть была обучена. Очевидно, что потребуется столько сетей Хопфильда, сколько определено элементов структуры, что несколько снизит скорость сопоставления (до O(N*N_e), где N_e – количество элементов структуры).

Вторым решением будет применение самообучающейся сети Кохонена [6],[7], представленная на рис. 2.

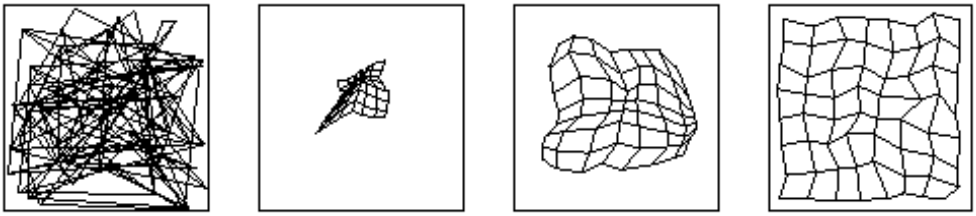

Рис. 2 Нейронная сеть Кохонена согласно [4, 65]
а – необученная
б – после 200-й итерации обучения
в – после 600-й итерации обучения
г – после 1900-й итерации обучения

В начале своего существования топология сети будет представляться множеством нейронов, соединенных между собой случайным образом и случайным-же образом активирующихся при подаче предложений на вход. После начального обучения сети нейроны будут организованы таким образом, что при подаче предложений, принадлежащих одной структуре, будут активироваться (выдавать сигнал “1”) только нейроны из некоторой, вполне определенной пространственной области. Таким образом, по принадлежности активированных нейронов областям будет определяться принадлежность предложения элементу структуры. Побочным, но, несомненно, полезным, эффектом будет самообучаемость сети, которая будет модифицироваться при поступлении похожих предложений, не входящих в обучающее множество.

Оба обозначенных решения представляют несомненный интерес, однако оставляют обширное поле для дальнейших исследований. Так, нерешенным остался вопрос цифрового представления предложений, подаваемых на вход сети. Одним из возможных путей оцифровки предложения представляется группа методов, используемых конвертерами текст-голос, где текст аппроксимируется периодической функцией. Еще одним важным вопросом является изучение адекватности той “эвристики”, которая будет присуща обученной сети при сопоставлении предложений, не входящих в обучающее множество. Такой анализ, однако, видится возможным в данное время, только экспериментальным.

Литература

1. Хопкрофт Дж., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. - М.: Издательский дом “Вильямс”. 2002. - 528 с.
2. Бархударов Л.С. Структура простого предложения современного английского языка. М.: Издательство “Высшая школа”. 1966. - 200с.
3. Sleator D.D., Temperley D. Parsing English with a Link Grammar. - Pittsburg: School of Computer Science, Carnegie Melon University. - 1993. - 14p.
4. Krose B., P. van der Smagt An introduction to Neural Networks. - Amsterdam: The University of Amsterdam. - 1996. - 135p.
5. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities //Proceedings of the National Academy of Sciences, №79. - 1982. - pp.2554-2558.
6. Kohonen T. Self-organized formation of topologically correct feature maps //Biological Cybernetics, №43. - 1982. - pp.59-69.
7. Kohonen T. Self-Organization and Associative Memory. Berlin: Springer-Verlag.