УДК 004.81

Анализ зависимостей текста на естественном языке с помощью BiLSTM-сетей

Артем Чернышов
аспирант кафедры 22, НИЯУ МИФИ, Москва, Россия
zexirius@gmail.com

Аннотация. В данной статье рассматривается идея проведения многоступенчатого процесса построения поискового образа запроса на естественном языке для использования в системе семантического поиска. Современные методы и инструменты обработки естественного языка широко используются в области машинного перевода. Исследования в области поисковых систем и семантического поиска в основном сосредоточены на хранении данных и дальнейшем анализе. Большинство поисковых систем используют огромное количество ранее накопленных пользовательских запросов для прогнозирования результатов поиска, не принимая во внимание это намерение пользователя путем качественной обработки запроса. Предлагаемый подход основан на выделении максимального количества информации из исходного запроса путем проведения синтаксического и семантического анализа, а также применении приемов синонимичного расширения. В данной статье описывается первый этап процесса построения модели поискового запроса, основанный на выделении синтаксических зависимостей из исходного предложения.

Ключевые слова: поиск, запрос, зависимости, нейронная сеть.

BiLSTM-based Approach to the Natural Language Text Dependencies Analysis

Artem Chernyshov
postgraduate student of the Department 22, National research nuclear University MEPhI, Moscow, Russia
zexirius@gmail.com

Abstract. This article discusses the idea of conducting a multistage process of building a search image of a query in natural language for use in the semantic search system. Modern methods and tools for processing natural language are widely used in the field of machine translation. Research on search engines and semantic search mainly focuses on data storage and further analysis. Most search engines use a huge amount of previously accumulated user queries to predict search results, without taking into account the user's intention through quality query processing. The proposed approach is based on the selection of the maximum amount of information from the original request by means of syntactic and semantic analysis, as well as the use of synonymous extension techniques. This article describes the first step in the process of building a search query model, based on extracting syntactic dependencies from the original sentence.

Keywords: search, query, dependencies, neural network.

DOI: 10.31432/1994-2443-2019-14-1-44-47

Цитирование публикации: Чернышов А. Анализ зависимостей текста на естественном языке с помощью BiLSTM-сетей // Информация и инновации. 2019.Т.14, № 1. С. ___-____.
Citation: Chernyshov A. BiLSTM-based approach to the natural language text dependencies analysis // Information and Innovations 2019.T.14, № 1.S. ___-____.

Введение

На данный момент, большинство современных поисковых систем работают по принципу выделения ключевых слов из поискового запроса и дальнейшего сопоставления с существующим поисковым индексом. Этот подход не учитывает семантику и отношения между словами в исходном запросе. Потеря скрытых зависимостей может привести к плохой релевантности результатов поиска. Тем не менее, данный подход имеет место, в том случае, когда имеющийся поисковый индекс огромен, а поисковая система руководствуется также профилем пользователя, который составляется сторонними системами.

Работая в строго определенной области, например, внутри конкретной организации, данный подход не выглядит состоятельным, поскольку огромный индекс отсутствует, а профиль пользователя никак не определен. Таким образом, работая над системой семантического поиска, мы не можем игнорировать семантику, поскольку использование простых ключевых слов приведет к низкой релевантности поисковой выдачи.

В рамках данной работы предлагается использование доменных онтологий для хранения поискового индекса и методы обработки естественного языка для построения иерархического изображения поискового запроса с сохранением семантических зависимостей для сопоставления и поиска на графе онтологий. Одной из задач является разработка адекватной модели поискового запроса, то есть поискового образа запроса, для дальнейшего использования в системе семантического поиска [1]. Для этого необходимо решить ряд проблем, связанных с зависимостями между словами на уровне фраз, проблемами определения синонимов контексте исходных слов и фраз и мерами их близости, определения контекста поиска и поисковых намерений пользователя в контексте области поиска [2].

В данной статье рассматривается одна из перечисленных выше проблем, связанных с зависимостями между словами, также представлен разработанный алгоритм, разработанный на основе классического разбора зависимостей.

Модели и методы

Один из способов выделения зависимостей внутри фразы — построения дерева зависимостей [1]. Здесь может быть два варианта — выделение зависимостей на уровне слов и на уровне фраз. Зависимости на уровне фраз хорошо подходят для понимания структуры предложения, для того чтобы понимать структуру конкретной фразы внутри предложения наиболее подходит грамматики на основе зависимостей.

В синтаксическом дереве зависимостей корнем как правило является глагол или любое другое слово, обозначающее действие. На втором ярусе, как правило, находятся объект и субъект действия, на более низких ярусах — характеристики объекта и субъекта и различные ограничители (время, количество, свойства и т.д.).

Все связи в дереве являются помеченными специальными POS-метками. Исходя из типа метки, можно отбрасывать незначащие термы, например, тег CASE объединяет предлоги, союзы и другие незначимые части речи, которые могут быть отброшены из рассмотрения.

Таким образом, поисковому запросу можно легко сопоставить иерархическую структуру с явно выделенными зависимостями между значащими термами.

Для построения соответствующей модели используется анализатор shift-reduce [1]. Для обучения используется рекуррентная нейронная сеть LSTM с одним скрытым слоем. В качестве данных для обучения используется заранее размеченный корпус русскоязычных текстов. В качестве функции потерь используем кросс-энтропию: $\mathcal{L}(\sigma) = -\sum_{i,j} \log(y_{ij})$ с L2 регуляризацией.

Конечная цель — получить последовательность действий — shift (left-arc, right-arc), reduce, по которой можно восстановить древовидную структуру.

Классическая архитектура нейронной сети LSTM описывается набором следующих формул:

$f_t = \sigma_g(W_t x_t + U_f h_{t-1} + b_f)$ — *вектор активации вектора забывания*

$i_t = \sigma_g(W_t x_t + U_i h_{t-1} + b_i)$ — *вектор активации входа*

$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$ — *вектор активации выхода*

$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$— *вектор состояния*

$h_t = o_t \circ \sigma_h(c_t)$ — *выходной вектор*

Для оценки полученной модели были использованы меры precision, recall и среднегармоническая F-мера.

Получаемую на выходе модель можно представить в виде графа:

$$G =< X, A >, \text{где}$$

- $X = \{ x_1, x_2, \dots, x_n \}$ — слова входного поискового запроса
- $A = \{ a_1, a_2, \dots, a_n \}$ — построенные связи между словами вида $a_i := x_i \underset{pos}{\longrightarrow} x_j$

Полученную модель можно использовать для поиска по онтологии. Формально онтологию можно представить в виде

$$O =< I, \mathrm{C}, \mathrm{P}, \mathrm{O} >, \text{где}$$

- $I = \{ o_1, o_2, \dots, o_n \}$ — объекты, или сущности предметной области
- $C = \{ c_1, c_2, \dots, c_n \}$ — классы,
- $P = \{ p_1, p_2, \dots, p_n \}$ — характеристики классов
- $O = \{ o_1, o_2, \dots, o_n \}$ — операторы между классами

Первым шагом поискового алгоритма является сопоставление корня исходного дерева с типом связи на онтологии. Так как связи в онтологии являются однозначно именованными, тогда как в исходном запросе они могут сильно различаться. Для того чтобы сопоставить корень с типом связи необходимо определить степень похожести между двумя словами — имя связи на графе и корень дерева.

$$sim = \min_{i=1,n}\left(\frac{\sum_{v=1}^{k} x_{iv}o_{i,v}}{\sqrt{\sum_{v=1}^{k} x_{iv}^2}\sqrt{\sum_{v=1}^{k} o_{iv}^2}}, \frac{\sum_{v=1}^{k} x_{iv}c_{i,v}}{\sqrt{\sum_{v=1}^{k} x_{iv}^2}\sqrt{\sum_{v=1}^{k} c_{iv}^2}}\right)$$

Для этого используется поиск по векторному пространству связей, для определения меры похожести используется косинусная мера. Как только найдена наиболее правдоподобная связь в онтологии, необходимо найти два типа вершин, которые она связывает [5]. Для этого также применяется поиск похожих слов на графе для вершин типа субъект и объект действия. Как только необходимая связь найдена, определяются условия в зависимости от типа связи. Это могут быть временные условия, локационные или характеристические.

Результаты

Для оценки модели, полученной в результате работы синтаксического анализатора, используются классические метрики — точность (precision), полнота (recall) и F-мера.

$$Rec = \frac{|S_{train} \cap S_{parsed}|}{|S_{train}|}$$

$$Prec = \frac{|S_{train} \cap S_{parsed}|}{|S_{parsed}|}$$

$$F = 2\frac{prec \times rec}{prec + rec}$$

Помимо стандартных метрик рассматриваются также дополнительные меры оценки точности анализатора. Labeled precision — относительное количество правильно распознанных пар-меток узлов в выходных данных анализатора. Labeled recall — относительное количество меток узлов-пар из обучающей выборки, получаемое в результате работы анализатора. Данные метрики используются для расчета таких мер как LAS (labeled attachement score) — относительного количество правильно определенных пар с учетом синтаксической метки, и UAS (unlabeled attachement score) — относительного количество правильно определенных пар без учета синтаксической метки. Помимо перечисленных выше метрик, также используется метод leaf-ancestor (LA), который сравнивает путь между узлами (от конечных узлов до корня) выходных данных синтаксического анализатора и путь в соответствующем предложении в тестовой выборке.. Сходство путей рассчитывается с помощью расстояния Левенштейна.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j), \min(i,j) = 0 \\ min(\left(lev_{a,b}(i-1,j)+1\right), \left(lev_{a,b}(i,j-1)+1\right), \left(lev_{a,b}(i-1,j-1)+1\right)) \end{cases}$$

Для обучения нейронной сети, являющейся ядром анализатора, мы использовали выборку предложений в формате CoNNL-U для русского языка — UD_Russian-SynTagRus. Он содержит 61889 предложений и 1106290 слов. В таблице 1 приводится результат расчета описанных выше метрик в сравнении с другими известными анализаторами.

В результате использования BiLSTM нейронной сети в качестве ядра анализатора, средняя точность распознавания зависимостей заметно увеличилась. Все перечисленные выше метрики показывают способность синтаксического анализатора правильно распознавать метки дуг и теги [6]. Мы также вычислили процент правильно распознанных тегов части речи и обнаружили, что разница между анализаторами незначительна. Например, точность SyntaxNet для способности распознавать теги ADJ, NOUN и VERB составляет 93,2450, 91,4477 и 83,7365 соответ-

Таблица 1.

	LAS		LA		UAS		F	
	train	test	train	test	train	test	train	Test
MALT	69,73	67,30	78,92	75,77	76,50	71,74	75,12	74,54
SyntaxNet	72,09	69,58	81,59	78,34	79,09	74,17	77,66	77,07
SNNDP	70,41	67,96	79,69	76,51	77,25	72,44	75,85	75,27
spaCy	68,07	65,70	77,04	73,97	74,68	70,04	73,33	72,77
Described parser	76,12	73,47	86,16	82,72	83,51	78,32	82,00	81,38

ственно, полученный анализатор — 92,3709, 90,8152 и 85,7920 соответственно.

Заключение

В данной статье описан первый этап многоступенчатого процесса обработки поисковых запросов. Данный процесс также содержит и другие шаги — расширение синонимов, распознавание контекста и пользовательских целей, распознавание именованных сущностей и общая проверка грамматики. Тем не менее, синтаксический анализ является важнейшим шагом в обработке естественного языка и в том числе обработке пользовательских запросов. Дальнейшие задачи включают разработку методов и алгоритмов извлечения пользовательских намерений с помощью семантического отображения картирования и синтаксического анализа, расширения синонимичных отношений с использованием векторных пространств слов.

ЛИТЕРАТУРА

1. Chernyshov A., Balandina A., Kostkina A., Klimov V. Intelligence Search Engine and Automatic Integration System for Web-Services and Cloud-Based Data Pro-viders Based on Semantics // Procedia Computer Science. 2016.

2. Kostkina A., Bodunkov D., Klimov V. Document Categorization Based on Usage of Features Reduction with Synonyms Clustering in Weak Semantic Map // Procedia Computer Science. 2018.

3. Kubler S., McDonald R., Nivre J. Dependency parsing // Synthesis Lectures on Human Language Technologies, Vol. 1, No. 1, 2009. pp. 1–127.

4. Nivre J. Transition-Based Parsing. 2013.

5. Chernyshov A., Balandina A., Klimov V. Intelligent Processing of Natural Language Search Queries Using Semantic Mapping for User Intention Extracting // Advances in Intelligent Systems and Computing. 2019.

6. Balandina A., Kostkina A., Chernyshov A., Klimov V. Dependency Parsing of Natural Russian Language with Usage of Semantic Mapping Approach // Procedia Computer Science. 2018.