Система ответов на вопросы, поддерживаемая извлечением информации

Рохини Шрихари Кимфони Инк.

5500 Главная улица

Уильямсвилл, Нью-Йорк 14221 rohini@cymfony.com

Абстрактный

В этом документе обсуждается система извлечения информации (IE), Textract, на естественном языке (NL) отвечает на вопросы (QA) и рассматривается роль IE в приложении QA. Он показывает: (i) тегирование именованных сущностей является важным компонентом для обеспечения качества, (ii) неглубокий парсер NL обеспечивает структурную основу для вопросов, и (iii) высокоуровневый независимый от домена IE может привести к прорыву QA.

Знакомство

С взрывным ростом информации в Интернете качество естественного языка признано как возможность с большим потенциалом. Традиционно QA привлекал многих исследователей Al, но большинство разработанных систем контроля качества - это игрушечные системы или игры, ограниченные лабораторией и очень ограниченной областью. Совсем недавно Конференция по поиску текста (TREC-8) разработала трек контроля качества, чтобы стимулировать исследования для применения в реальном мире.

Из-за небольшой лингвистической поддержки со стороны анализа текста обычные IR-системы или поисковые системы на самом деле не выполняют задачу поиска информации; на самом деле они нацелены только на поиск документов. Следующая цитата из QA Track Specifications (www.research.att.com/ —singhal/qa-track-spec.txt) в сообществе TREC иллюстрирует этот момент.

Современные информационно-поисковые системы позволяют нам находить документы, которые могут содержать соответствующую информацию, но большинство из них оставляют пользователю извлечение полезной информации из ранжированного списка. Это оставляет (часто Вэй Ли)

Кимфони Инк.

5500 Главная улица

Уильямсвилл, Нью-Йорк 14221 wei@cymfony.com

не желает) пользователь с относительно большим количеством текста для потребления. Существует острая потребность в инструментах, которые уменьшили бы объем текста, который, возможно, придется читать для получения желаемой информации. Этот трек направлен на то, чтобы сделать именно это для специального (и популярного) класса поведения поиска информации: ОТВЕТЫ НА ВОПРОСЫ. У людей есть вопросы, и им нужны ответы, а не документы. Автоматические ответы на вопросы, безусловно, станут значительным шагом вперед в современной технологии поиска информации.

Kupiec (1993) представил систему контроля качества MURAX с использованием онлайн-энциклопедии. Эта система использовала технологию надежного мелкого синтаксического анализа, но страдала от отсутствия базовой поддержки извлечения информации. Фактически, наиболее сигинифантное продвижение IE, а именно технология NE (Named Entity), произошло после Kupiec (1993), благодаря программе MUC (MUCO 1998). Высокоуровневая технология IE за пределами NE до недавнего времени не находилась в стадии возможного применения.

AskJeeves запустил портал QA (www.askjeeves.com). Он оснащен довольно сложным парсером вопросов на естественном языке, но не дает прямых ответов на заданные вопросы. Вместо этого он направляет пользователя на соответствующие веб-страницы, как это делает традиционная поисковая система. В этом смысле AskJeeves выполнил только половину работы по обеспечению качества.

|  |
| --- |
| \* Эта работа была частично поддержана грантами SBIR F30602-98-C-0043 и F30602-99-C-0102 от Исследовательской лаборатории ВВС (AFRL) / IFED. |

Мы считаем, что QA является идеальным испытательным стендом для демонстрации возможностей IE. Существует естественное сотрудничество между IE и IR; мы рассматриваем QA как один из основных интеллектов, который IE может предложить IR.

Таким образом, важный вопрос заключается в том, какой тип IE может поддерживать IR в QA и насколько хорошо он его поддерживает? Это составляет основную тему данной статьи. Мы структурируем оставшуюся часть документа следующим образом. В разделе 1 мы сначала дадим обзор базовой технологии IE, которую разрабатывает наша организация. В разделе 2 рассматривается система контроля качества. В разделе 3 описываются ограничения нынешней системы. Наконец, в разделе 4 мы предлагаем более сложную систему контроля качества, поддерживаемую тремя уровнями IE.

1 Обзор Textract IE

В последнее десятилетие наблюдается большой прогресс и интерес в области IE. В США спонсируемая DARPA программа Tipster Text Program [Grishman 1997] и Конференции по пониманию сообщений (MUC) [MUC-7 1998] были движущей силой для разработки этой технологии. Фактически, спецификации MUC для различных задач IE стали де-факто стандартами в исследовательском сообществе IE. Поэтому необходимо представить наши усилия в области ИП в контексте программы MUC.

MUC делит IE на отдельные задачи, а именно: NE (Named Entity), TE (Template Element), TR (Template Relation), CO (Co-reference) и ST (Scenario Templates) [Chinchor & Marsh 1998]. Наше предложение по трем уровням IE смоделировано по стандартам MUC с использованием представления в стиле MUC. Однако мы модифицировали определения задач MUC IE, чтобы сделать их более полезными и практичными. Точнее, мы предлагаем иерархическую, 3-уровневую архитектуру для разработки системы IE ядра, которая во всем является независимой от домена.

Ядром этой системы является современный теггер NE [Srihari 1998], названный Textract 1.0, Теггер Textract NE достиг скорости и точности, сопоставимых с немногими развернутыми системами NE, такими как NetOwl [Krupka & Hausman 1998] и Nymble [Bikel et al 1997].

Следует отметить, что в нашем определении NE мы значительно расширили тип информации, подлежащей извлечению. В дополнение ко всем определенным MUC типам NE (человек, организация, местоположение, время, дата, деньги и процент), следующие типы / подтипы информации также идентифицируются модулем TextractNE:

 продолжительность, частота, возрастной номер, дробь, десятичный, порядковый номер, вес математического уравнения, длина, температура, угол, площадь, емкость, скорость, рейтинг продукта, программный адрес, электронная почта, телефон, факс, телекс, www имя (имя собственное по умолчанию)

Также идентифицируется информация подтипа, такая как компания, государственное учреждение, школа (принадлежащая к типовой организации) и военное лицо, религиозное лицо (принадлежащее лицу). Эти новые подтипы обеспечивают лучшую основу для определения множественных связей между идентифицированными сущностями и для поддержки функциональности ответов на вопросы. Например, ключом к обработчику вопросов является определение запрашивающей точки (кто, что, когда, где и т. Д.). Во многих случаях запрашиваемая точка соответствует NE за пределами определения MUC, например, вопросы «как+прилагательное»: как долго (продолжительность или длина), как далеко (длина), как часто (частота), сколько лет (возраст) и т. Д.

IE уровня 2, или CE (Correlated Entity), занимается извлечением предопределенных множественных связей между сущностями. Рассмотрим в качестве примера физическое лицо; Прототип TextractCE способен извлекать ключевые отношения, такие как возраст, пол, принадлежность, должность, birth\_time, birth\_place, супруг, родители, дети, where\_from, адрес, телефон, факс, электронная почта, дескрипторы. Как видно, информация в СЕ представляет собой мини-резюме или профиль организации. В целом, шаблон CE интегрирует и значительно обогащает информацию, содержащуюся в MUC TE и TR.

Конечной целью наших усилий в IE является дальнейшее извлечение открытых общих событий (GE или IE уровня 3) для получения информации о том, кто что (кому) делал, когда (или как часто) и где. Под общими событиями мы относимся к структурам аргументов, сосредоточенным вокруг понятий глаголов плюс связанная с ними информация о времени/частоте и местоположении. Мы показываем пример нашего определения GE, извлеченный из текста ниже:

Джулиан Хилл, химик-исследователь, чье случайное открытие жесткого, похожего на таффи соединения произвело революцию в повседневной жизни после того, как оно доказало свою ценность в войне и ухаживании,

умер в воскресенье в Хокессине, штат Дель.



|  |  |
| --- | --- |
| ПРЕДИКАТ: | тем |
| АРГУМЕНТЫ: | Джулиан Хилл |
| ВРЕМЯ: | Воскресенье |
| МЕСТОПОЛОЖЕНИЕ: | Хокессин, Дель |

лучшие из обеих парадигм при преодолении их соответствующих слабостей [Srihari 1998, Li & Srihari 2000].

2 QA с поддержкой NE

В этом разделе представлена система контроля качества, основанная на тегировании именованных сущностей. Из 200 вопросов, которые составляли соревнование TRECÆ QA track, более 80% задавали NE, например, кто (PERSON), когда (TIME I DATE), где (LOCATION), как далеко (LENGTH). Таким образом, было доказано, что теггер NE очень полезен. Конечно, NE целевого типа только необходим, но не полный для ответа на такие вопросы, потому что NE по своей природе извлекает из текста только изолированные отдельные сущности. Тем не менее, используя даже грубые методы, такие как «ближайший NE к запрашиваемым ключевым словам» или «NE и связанные с ним ключевые слова в одной строке (или том же абзаце и т. Д.)», в большинстве случаев система контроля качества смогла извлечь текстовые части, которые содержали ответы в первой пятерке списка.

На рисунке 2 показана системная конструкция прототипа TextractQA. Существует два компонента для прототипа QA: обработчик вопросов и текстовый процессор. Модуль Text Matcher связывает два результата обработки и пытается найти ответы на обработанный вопрос. Сопоставление основано на ключевых словах, а также на типе NE и их общем расположении в одном предложении.

|  |
| --- |
| Обработчик вопросов  Ответ |

Рисунок 2: Архитектура прототипа Textract/QA 1.0

Общий алгоритм ответа на вопрос выглядит следующим образом:

На рисунке I показана общая архитектура системы IE Textract, которую разрабатывает наша организация.

|  |
| --- |
| Модули ядра IEЛингвистические модули    Прикладные модули  NE: N•smd Энтай Taggi8QA: Ответ-m  CE: Corr&ed EntiyBR: Intcll\*nt Browsig  GE: Gemr.l Eve.u ЭкстратонАС; Сумрруризбтку•  CO: Знаки со-ссылок |

Рисунок 1: Архитектура системы Textract IE

Ядро системы состоит из трех модулей ядра IE и шести лингвистических модулей. Многоуровневые лингвистические модули служат базовой системой поддержки для различных уровней IE. Результаты IE хранятся в базе данных, которая является основой для приложений, связанных с IE, таких как QA, BR (просмотр, многопоточность и визуализация) и AS (автоматическое суммирование). Подход к IE, принятый здесь, состоит из уникального сочетания машинного обучения и FST (преобразователь конечного состояния) системы, основанной на правилах [Roche & Schabes 1997]. Объединяя машинное обучение с системой на основе правил FST, мы можем использовать

Вопрос о процессе

Вопрос о мелком разборе

Определение точки запроса

Расширение вопроса (с использованием списков слов)

Процессуальные документы

Токенизация, POS-тегирование, индексирование NE

Мелкий синтаксический анализ (еще не используется)

Совпадения текста

Пересечение результатов поисковой системы с ответами на рейтинг NE

2.1 Обработка вопросов

Результаты обработки вопросов представляют собой список ключевых слов плюс информацию для постановки вопроса. Например, вопрос:

1. Кто получил Нобелевскую премию мира 1998 года?

содержит следующие ключевые слова: вон, 1998, Нобелевская премия, Мир, Премия. Вопросительный пункт Кто относится к человеку типа NE. Выходные данные перед расширением вопроса представляют собой простой шаблон из 2 функций, как показано ниже:

1. asking\_point: ПЕРСОНА key\_word:{won, 1998, Нобелевская премия,

Мир, Премия }

Ниже приведен пример, где запрашиваемая точка не соответствует какому-либо типу NE в нашем определении.

1. Почему Дэвид Кореш попросил у ФБР текстовый процессор?

Затем система сопоставляет его со следующим шаблоном вопроса:

1. asking\_point: ПРИЧИНА key\_word:{ спросите, Давид, Кореш,

ФБР, текст, процессор }

Обработчик вопросов сканирует вопрос для поиска вопросительных слов (wh-слов) и сопоставляет их с соответствующими типами / подтипами NE или заранее определенными понятиями, такими как REASON.

Для этой цели мы принимаем два набора правил сопоставления шаблонов: (i) правила сопоставления шаблонов на основе структуры; ii) простые правила сопоставления шаблонов на основе ключевых слов (рассматриваемые в качестве правил по умолчанию).

Довольно легко исчерпать второй набор правил, так как вопросительные вопросительные слова/фразы образуют замкнутый набор. Для сравнения, разработка первого свода правил постоянно совершенствуется и расширяется. Такая стратегия использования двух наборов правил приводит к надежности обработчика вопросов.

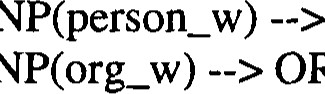
Первый набор правил основан на неглубоком анализе результатов вопросов, используя Cymfony FST на основе Shallow Parser. Этот парсер идентифицирует основные синтаксические конструкции, такие как BaseNP (базовая фраза существительного), BasePP (базовая предложная фраза) и VG (группа глаголов).

Ниже приведен пример первого набора правил.

1. Название NP (город I страна I компания)

СИТИСТРАНАКОМПАНИЯ

1. Имя ПЕРСОНА

 [81 Название ОРГАНИЗАЦИИ

[9] Имя NP (НЕ person\_w, НЕ org\_w)

ИМЯ

Правило [6] проверяет головное слово НП. Он охватывает такие случаи, как VG[Namel NP[страна], что VG [разрабатывает] NP [магнитная левитационная железнодорожная система]. Правило [71 работает для таких случаев, как VG[Name] NP[первый частный гражданин] VG[летающий] PP[в космосе], поскольку гражданин принадлежит к слову класс person\_w. Правило [9] является универсальным правилом: если NP не относится к классу человека (person\_w) или организации (org\_w), то запрашивающей точкой является имя собственное (по умолчанию NE), часто реализуется в английском языке в заглавной строке слов. В качестве примера можно привести фильм «Назовите», который выиграл «Золотого медведя» на Берлинском кинофестивале.

Списки слов org\_w и person\_w в настоящее время ведутся вручную на основе проверки больших объемов текста. В настоящее время предпринимаются усилия по автоматизации изучения таких списков слов с использованием методов машинного обучения.

Мы использовали следующие преобразования шаблонов для расширения нашего набора правил:

(Пожалуйста) назовите NP[X]

--> что/что Aux(be) (название) NP[X] --> NP(что/что... X)

Другими словами, четыре правила расширены до 12 правил. Например, правило [10] ниже соответствует правилу [6]; Правило [I l] является производным от Правила

[10] что/какой Aux(be) NP (город I страна I компания)

ГОРОД I СТРАНА I КОМПАНИЯ

[l IJ NP(что/что ... person\_w)

ЧЕЛОВЕК

Правило [10] извлекает вопросительный момент из таких случаев, как NP[What] Aux[is] NP[крупнейшая страна] PP[в мире]. Правило [l l] охватывает следующие вопросы: NP[Какой художник по костюмам] VG[решил], что NP[Майкл Джексон] VG[должен носить только] NP[одну перчатку], NP[Какой бывший член Ку-клукс-клана]

VG[выиграл] NP[выборная должность] PP[в США], NP[Какой нобелевский лауреат] VG[был исключен] PP[с Филиппин] PP[до конференции] PP[по Восточному Тимору], NP[Какой известный коммунистический лидер] VG[умер] PP[в Мехико] и т.д.

Как видно, неглубокий синтаксический анализ помогает нам захватывать различные выражения вопросов естественного языка. Тем не менее, есть случаи, когда некоторого простого сопоставления шаблонов на основе ключевых слов было бы достаточно, чтобы уловить вопросительную точку. Это наш второй свод правил. Эти правила используются, когда первый набор правил не дает результатов. Ниже приведен пример таких правил:

1. кто/кто ПЕРСОНА
2. когда ВРЕМЯ/ДАТА
3. где/какое место РАСПОЛОЖЕНИЕ
4. какое время (суток) --> ВРЕМЯ
5. какой день (недели) --> ДЕНЬ
6. что]какой месяц МЕСЯЦ
7. какой возраст/сколько лет --> AGE
8. какой бренд ПРОДУКТ

[201 какое ИМЯ

[21] Как далеко/высокая/высокая ДЛИНА

[221 насколько большая/большая/маленькая ПЛОЩАДЬ

1. насколько тяжел ВЕС
2. как богаты ДЕНЬГИ
3. как часто ЧАСТОТА
4. сколько НОМЕР
5. как долго ДЛИНА/ПРОДОЛЖИТЕЛЬНОСТЬ [28] почему/по какой ПРИЧИНЕ

На этапе расширения вопроса шаблон в [4] будет расширен до шаблона, показанного в [29]:

|  |  |
| --- | --- |
| [29] asking\_point: | {becauselbecause on due tolthanks tolsincel in orderlto VB ) |
| key\_word: | {asklaskslaskedlasking, |



слово, процессор)

Последний элемент в списке asking\_point пытается найти инфинитив, проверяя слово, за которым следует глагол (с тегом части речи VB). Как известно, инфинитивные глагольные фразы часто используются в английском языке для объяснения причины какого-либо действия.

* 1. Обработка текста

Что касается обработки текста, мы сначала отправляем вопрос непосредственно в поисковую систему, чтобы сузить пул документов до первого n, скажем, 200, документов для обработки IE. В настоящее время это включает в себя токенизацию, POS-тегирование и NE-тегирование. Будущие планы также включают несколько уровней синтаксического анализа; они необходимы для поддержки извлечения CE и GE. Следует отметить, что все эти операции чрезвычайно надежны и быстры, функции, необходимые для индексации большого объема текста. Синтаксический анализ осуществляется с помощью каскадных грамматик преобразователей конечных состояний.

* 1. Сопоставление текста

Средство сопоставления текста пытается сопоставить шаблон вопроса с обработанными документами как по запрашиваемой точке, так и по ключевым словам. Существует предварительный стандарт ранжирования, встроенный в матчер, чтобы найти наиболее вероятные ответы. Первичный рейтинг — это количество уникальных ключевых слов, содержащихся в предложении. Вторичный рейтинг основан на порядке, в котором ключевые слова появляются в предложении, по сравнению с их порядком в вопросе. Третий рейтинг основан на том, существует ли точное или вариантное соответствие для ключевого глагола.

В соревнованиях trec-8 QA точность Cymfony QA составила 66,0%. Учитывая, что мы использовали только технологию NE для поддержки QA в этом запуске, 66,0% - это очень обнадеживающий результат.

1. Ограничение

Первое ограничение связано с типами вопросов. В настоящее время обрабатываются только wh-вопросы, хотя планируется, что вопросы «да-нет» будут обработаны, как только мы представим шаблоны CE и GE для поддержки QA. Среди wh-вопросов, почему-вопрос и как-вопрос i являются более сложными, потому что запрашиваемая точка не может быть просто сопоставлена с типами / подтипами NE.

Второе ограничение связано с характером вопросов. На такие вопросы, как Где я могу найти домашнюю страницу лауреатов Оскара или Где можно найти информацию о произведениях Шекспира, может легко ответить система, основанная на хорошо поддерживаемой базе данных домашних страниц. Поскольку наша система основана на обработке основополагающих документов, правильный ответ не может быть предоставлен, если в обрабатываемых документах нет такого ответа (явно выраженного на английском языке). В TREC-8 QA это не проблема, так как каждый вопрос гарантированно имеет хотя бы один ответ в данном пуле документов. Однако в реальном сценарии, таком как портал обеспечения качества, предполагается, что результаты IE, основанные на обработке документов, должны дополняться другими источниками знаний, такими как электронная копия желтых страниц или другие вручную поддерживаемые и обновленные базы данных.[[1]](#footnote-1)

Третьим ограничением является отсутствие лингвистической обработки, такой как разбор на уровне предложений и перекрестная сенденциальная совместная ссылка (CO). Эта проблема будет постепенно решаться, когда в систему будет внедрена высокоуровневая технология IE.

1. Будущая работа: Многоуровневый контроль качества, поддерживаемый IE

В настоящее время разрабатывается новая архитектура контроля качества; он будет использовать все уровни системы IE, включая CE и GE.

Первый вопрос заключается в том, насколько CE может способствовать лучшей поддержке QA. Обнаружено, что есть некоторые часто встречающиеся вопросы, на которые можно лучше ответить, как только будет предоставлена информация CE. Эти вопросы бывают двух типов: (i) что/кто спрашивает о NE; ii) вопросы взаимоотношений.

Вопросы следующего формата требуют шаблонов CE в качестве лучших ответов: кто / что такое NE? Например, кто такой Джулиан Хилл? Кто такой Билл Клинтон? Что такое Дюпон? Что такое Кимфони? Чтобы ответить на эти вопросы, система может просто получить соответствующий шаблон CE, чтобы предоставить «собранный» ответ, как показано ниже.

|  |  |
| --- | --- |
| Q: | Кто такой Джулиан Хилл? |
| A: | имя:Джулиан Вернер Хилл тип:ЧЕЛОВЕК возраст:91 пол:МУЖЧИНА должность:исследователь химик принадлежность:Du Pont co.  образование: Вашингтонский университет;  С |
| Q: | Что такое Дюпон? |
| A: | имя: Du Pont Co. тип: ПЕРСОНАЛ КОМПАНИИ: Джулиан Хилл; Уоллес Карозерс. |

Вопросы конкретно об отношениях CE включают: В какой компании работал Джулиан Хилл? (аффилиация) Кто является сотрудниками Du Pont Co. ? (взаимоотношения с персоналом) Что делает Джулиан Хилл? (отношения между должностью и профессией) Какой университет окончил Джулиан Хилл? (образовательные отношения) и т.д. [[2]](#footnote-2)

Следующим вопросом являются отношения между GE и QA. Мы верим, что технология GE приведет к прорыву в области контроля качества.

Чтобы извлечь шаблоны GE, текст проходит серию лингвистической обработки, как показано на рисунке 1. Следует отметить, что обработка вопросов предназначена для прохождения параллельных процессов и совместного использования одних и тех же ресурсов НЛП до момента сопоставления и ранжирования.

Слияние шаблонов вопросов и шаблонов GE в Template Matcher довольно простое. Поскольку они оба подвергаются одной и той же обработке НЛП, результирующие семантические шаблоны имеют одинаковую форму. Как шаблоны вопросов, так и шаблоны GE соответствуют довольно стандартным/предсказуемым шаблонам (значение PREDICATE является открытым, но структура остается стабильной). Точнее, пользователь может задавать вопросы о самих общих событиях (сделал что) и/или об участниках мероприятия (кто, кто, что) и/или о времени, частоте и месте событий (когда, как часто, где). Это касается, безусловно, большинства типов общих вопросов потенциального пользователя.

Например, если пользователь заинтересован в событиях приобретения компании, он может задать такие вопросы, как: Какие компании приобретены Microsoft в 1999 году? Какие компании Microsoft приобрела в 1999 году? Затем наша система разберет эти вопросы в шаблоны, как показано ниже:

[31]

ПРЕДИКАТ: приобрести

АРГУМЕНТЫ : Майкрософт

АРГУМЕНТ2: КАКОЕ ВРЕМЯ: 1999

Если пользователь хочет знать, когда произошло какое-то приобретение, он может спросить: когда был приобретен Netscape? Затем наша система переведет его в шаблон ниже:

[32]

ПРЕДИКАТ: приобрести

АРГУМЕНТЫ: ВОЗ

АРГУМЕНТ2: ВРЕМЯ NETSCAPE: КОГДА

Обратите внимание, что ВОЗ, ЧТО, КОГДА выше являются переменными, экземпляры которых необходимо создать. Такие шаблоны вопросов служат ограничениями поиска для фильтрации событий в нашей извлеченной базе данных шаблонов GE. Поскольку шаблоны вопросов и извлеченный шаблон GE имеют одну и ту же структуру, достаточно простой операции слияния. Тем не менее, есть два важных вопроса, на которые необходимо ответить: (i) что, если в вопросе используется другой глагол с тем же значением, чем тот, который используется в обработанном тексте? (ii) что делать, если вопрос задается о чем-то, выходящем за рамки информации GE (или CE)? Это вопросы, которые мы сейчас исследуем.

Ссылки

Bikel D.M. et al. (1997) Nymble: a High-Pefformance Learning Name-finder. «Труды пятой конференции по прикладной обработке естественного языка», издательство Morgan Kaufmann, стр. 194-201

Чинчор Н. и Марш Э. (1998) MUC-7 Определение задачи извлечения информации (версия 5. l), "Труды MUC-T'.

Гришман Р. (1997) ТИПСТЕР Архитектурный дизайн

Версия документа 2.3. Технический отчет, DARPA

Krupka G.R. and Hausman K. (1998) IsoQuest Inc.: Описание системы экстракторов NetOwl (TM), используемой для MUC-7, "Proceedings of MUC-7".

Kupiec J. (1993) MURAX.• Надежный лингвистический подход к ответам на вопросы с использованием on-line Encyclopaedia, "Proceedings of SIGIR-93 93" Pittsburgh, Penna.

Li, W & Srihari, R. 2000. Гибкий алгоритм обучения извлечению информации, конечный технический

Доклад, Исследовательская лаборатория ВВС, Римский исследовательский сайт, Нью-Йорк

MUCO (1998) Труды седьмой Конференции по пониманию посланий (MUC-7), опубликованные на веб-сайте http://www.muc.saic.com/

Рош Э. и Шабес Ю. (1997) Конечное состояние

Language Processing, MIT Press, Cambridge, MA Srihari R. (1998) A Domain Independent Event

Инструментарий для извлечения, AFRL-1F-RS-TR-1998-152 Заключительный технический отчет, Исследования ВВС

Лаборатория, Римская исследовательская площадка, Нью-Йорк

1. Например, Как сделать a шоколадный торт? Как+Вопросы прилагательных (например, как долго, как большой сколько лет и т.д.) aРе обработано довольно хорошо. [↑](#footnote-ref-1)
2. Ан Поддержка альфа-версии TextractQA около как NE, так и CE были внедрены и тестируются. [↑](#footnote-ref-2)