# Описание алгоритма.

Алгоритм основан на чисто инженерном подходе. Для извлечения фактов в ходе синтаксического анализа текст сравнивается с лексико-синтаксическими шаблонами, составленными на основе примеров текстов, содержащих факты искомого типа. При соответствии текста шаблону его соответствующие элементы нормализуются и заносятся в поля факта.

Извлекается факт Occupation, имеющий следующие атрибуты:

* Who: физическое лицо
* Where: организация, в которой это лицо работает
* Position: должность лица в этой организации

Используемые шаблоны основаны на наиболее частой форме изложения факта Occupation, а именно использовании описательной именной группы, содержащей должность и место работы человека, непосредственно рядом с его именем. Шаблоны учитывают, что не все такие группы соответствуют факту Occupation, а также что они могут следовать до или после имён, в том числе и при перечислении.

Шаблоны извлечения имён организаций-работодателей ориентированы в основном на политические и административные организации, с сильной опорой на газеттир.

# Описание программы.

В качестве основной программы используется Томита-парсер. Встроенные средства Томита-парсера осуществляют графематический и морфологический анализ.

Томита-парсер также содержит встроенный алгоритм для синтаксического анализа, но не содержит грамматик. Соответственно, для извлечения фактов необходимо добавить грамматики, по которым производится синтаксический анализ и интерпретация факта, а также словари ключевых слов и многословных сущностей. Грамматики аналогичны контекстно-свободным, но расширены проверкой и согласованием характеристик слов. Синтаксическая омонимия разрешается с использованием покрытий (длина фрагмента, для которого построен вывод по шаблону) и весов деревьев разбора, выводимых из весов отдельных правил.

Для извлечения факта «Occupation» в словарь были включены следующие категории ключевых слов:

* Должности. Ключевое слово этой категории обязательно содержится в шаблоне и заполняет слот Position фрэйма факта
* Формы организаций. Служат для расшифровки имён компаний, приведённых в кавычках.
* Территориально-правовые образования (города, государства и т.п.). Служат для определения властных структур как мест работы.

Конкретных типов ключевых слов в газеттире больше, так как каждая из вышеперечисленных категорий может выражаться с использованием разных синтаксических средств, что требует определения разных типов ключевых слов для извлечения аналогичных сущностей.

В ходе изначальной разработки шаблонов, заполнение словарей производилось в соответствии с примерами из корпуса.

Следует отметить, что использование Томита-парсера накладывает ограничения на возможности извлечения и интерпретации фактов. Основным непреодолимым ограничением является то, что сопоставление текста с шаблонами грамматики осуществляется только в пределах одного предложения, так что факты, не выраженные полностью в одном предложении, не могут быть выделены. Также имеются заметные проблемы при нормализации имён собственных, так как Томита-парсер не распознаёт их как специфическую грамматическую категорию. При этом слово, которое может быть распознано как форма обычного слова русского языка, нормализуется как это слово (имя «Сами» преобразуется в «сам», фамилия «Удальцов» – в «Удалец»).

# Описание методики тестирования.

Поскольку программа основана на чистом инженерном подходе, её работа не предусматривает машинного обучения. Таким образом, тестирование производится непосредственно путём запуска программы на примерах текстов, измерения характеристик её работы и индивидуального рассмотрения обнаруженных ошибок.

В связи с вышеупомянутыми проблемами нормализации, при тестировании использовалось как прямое сравнение полученных результатов с эталонными, так и вариант сравнения, засчитывавший разные словоформы одного слова как совпадение.

# Примеры работы.

Разбор списка. Отметим, что для Михаила Вельмакина выделены обе должности, а Лев Пономарёв не выделен вообще.

Пользователь ЖЖ zyalt сообщает , что среди задержанных оказался депутат муниципального собрания района Отрадное , сопредседатель московского отделения движения "Солидарность" Михаил Вельмакин , известный правозащитник Лев Пономарев , координатор движения "Левый фронт" Сергей Удальцов .

Occupation

{

Who = Вельмакин Михаил

Position = депутат

Where = муниципальное собрание района Отрадное

}

Occupation

{

Who = Вельмакин Михаил

Position = сопредседатель

Where = московское отделение движения "Солидарность"

}

Occupation

{

Who = Удалец Сергей

Position = координатор

Where = движение "Левый фронт"

}

Пример со списком мест работы для одного человека. Именная группа «известного борца с ксенофобией» не содержит ключевого слова для получения должности и не интерпретируется как часть факта Occupation, но входит в интерпретированный фрагмент. Также, слово «члена» соотнесено с обоими местами работы.

Инициатива "Яблока" оказалась неожиданной не только для Александра Белова, но и для его идейного оппонента Александра Брода , известного борца с ксенофобией , члена Общественной палаты и российско-американской рабочей группы по гражданскому обществу .

Occupation

{

Who = Брод Александр

Position = член

Where = Общественная палата

}

Occupation

{

Who = Брод Александр

Position = член

Where = российско-американская рабочая группа по гражданскому обществу

}

# Описание результатов.

Ввиду неполного газеттира были рассмотрены разные варианты фильтрации словаря правильных ответов. В качестве методов фильтрации были рассмотрены:

* Фильтрация по газетиру: в каждом поле, должно встречаться хотя бы одно слово из газеттира
* Фильтрация по сложности: не рассматривались ответы повышенной сложности
* Фильтрация по присутствию в одном предложении: наша система не может извлекать факты далее одного предложения
* Композиция вышеуказанных фильтров

В итоге получились следующие результаты:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Исходный сет | Отильтрованный по газеттиру | Отфильтрованный по сложности | Отфильтрованный по предложениям | Полностью отфильтрованный |
| devset - EQ | | **Ответов:** | | **75** | |
| Precision: | 0,13333333 | 0,133333333 | 0,133333333 | 0,08 | 0,08 |
| Recall: | 0,04761905 | 0,047619048 | 0,047619048 | 0,028571429 | 0,028571429 |
| Fmeasure: | 0,07017544 | 0,070175439 | 0,070175439 | 0,042105263 | 0,042105263 |
| devset - normalisation | |  | |  | |
| Precision: | 0,61333333 | 0,6 | 0,586666667 | 0,306666667 | 0,28 |
| Recall: | 0,21904762 | 0,214285714 | 0,20952381 | 0,10952381 | 0,1 |
| Fmeasure: | 0,32280702 | 0,315789474 | 0,30877193 | 0,161403509 | 0,147368421 |
| testset - EQ | | **Ответов:** | | **87** | |
| Precision: | 0,1954023 | 0,195402299 | 0,195402299 | 0,08045977 | 0,08045977 |
| Recall: | 0,03828829 | 0,038288288 | 0,038288288 | 0,015765766 | 0,015765766 |
| Fmeasure: | 0,06403013 | 0,064030132 | 0,064030132 | 0,026365348 | 0,026365348 |
| testset - normalisation | |  | |  | |
| Precision: | 0,56321839 | 0,551724138 | 0,563218391 | 0,252873563 | 0,24137931 |
| Recall: | 0,11036036 | 0,108108108 | 0,11036036 | 0,04954955 | 0,047297297 |
| Fmeasure: | 0,18455744 | 0,18079096 | 0,184557439 | 0,082862524 | 0,079096045 |

Соответсвенно devset и testset – наборы данных, EQ и normalisation – методы сравнения.

Заметно, что результаты ухудшаются с фильтрацией, это связано с тем, что мы удаляем подходящие нам факты, особенно после нормализации.

Особенно неожиданный результат при фильтрации «по предложениям», очевидно, что проблема опять же в нормализации слов, которая проводилась в рамках данного фильтрования, но видимо давала разные результаты. Также стоит обратить внимание, что иногда место обозначалось не существительным, а прилагательным, что уже даёт другую нормализацию.

# Анализ полученных результатов.

## Описание системы тестирования

Для проведения измерений был создан скрипт на языке Python 3.5, который:

1. Подготавливал конфигурационные файлы
2. Проводил параллельный запуск томиты на исходных текстах
3. Анализировал полученные результаты
   1. Сравнивая на идентичность извлечённые факты
   2. Сравнивая схожесть итоговых фактов (приводя к нормальной форме слова)
4. Анализировал полученные результаты с учётом невозможности существующим парсером извлечь какие-либо факты
   1. Не считая задания повышенной сложности
   2. Не считая факты, в которых не содержится схожих слов из газетира
   3. Не считая факты, которые невозможно извлечь из одного предложения
5. Формировал на основе результатов файл отчёта по каждому тесту с информацией об ошибках 4-х видов:
   1. Факты, найденные с недочётами
   2. Не найденные факты
   3. Факты, которые были найдены только нами
   4. Источники, из которых не было найдено ни одного факта
6. Считал Precision, Recall, F-меру

Схожими два факта считались, если при рассмотрении каждого поля, после нормализации информации из него, мы получали совпадение не менее 75% фразы. При выделении ошибок типа «a», рассматривались пары, в которые суммарно совпадали не менее чем на 2/3. Для оценки совпадения мы брали отношение: количество совпавших слов / количество слов в факте для тестирования.

Файлы проекта доступны: <https://github.com/nikmedoed/Extracting-entities-workers-PZKL-MSU-CMC-lv6/>

## Обзор ошибок и проблем

Благодаря наличию файла ошибок, мы можем рассмотреть в чём проблемы парсера.

В ходе отладки была выявления следующая ошибка:

Факты из источника book\_100 не совпали

Должно быть

Hard: F | a | l | s | e

Who: Вельмакин Михаил

Position: сопредседатель

Where: московское отделение движения Солидарность

получилось:

Who: Вельмакин Михаил

Position: сопредседатель

Where: московское отделение движения "Солидарность"

Факт не совпал только из-за кавычек, поэтому оценщик стал считать равными слова в кавычках и без при анализе типа «b». Кроме того, есть ещё разные виды кавычек, что тоже доставляет неудобств.

Ещё одна типичная ошибка

Факты из источника book\_349 не совпали

Должно быть

Who: Нетаньяху Биньямин

Hard: F | a | l | s | e

Position: глава

Where: правительство Израиля

получилось:

Who: Израиль Биньямин Нетаньяху

Position: глава

Where: правительство

Ещё один класс ошибок связан с двойственной трактовкой некоторых терминов в том числе в правильных ответах. Иногда в ответах предусмотрены варианты, т.е. для России может быть указано РФ | России | Российской федерации, но иногда этого не бывает и вот что получается:

Факты из источника book\_3966 не совпали

Должно быть

Position: президент

Hard: F | a | l | s | e

Where: РФ

Who: Путин Владимир

получилось:

Position: президент

Where: Россия

Who: Путин Владимир

Иногда встречается такая ошибка:

Факты из источника book\_3910 не совпали

Должно быть

Position: Министр иностранных дел

Hard: F | a | l | s | e

Where: Эстония

Who: Пентус Кейт

получилось:

Position: министр иностранных дел

Where: Эстония Кейт

Who: Роозиманнус Пентус

В предложении написано Министр иностранных дел Эстонии Кейт Пентус Роозиманнус, но парсер не разобрал и счёт часть имени частью страны, но интересно, что в правильно ответе вообще нет части имени.

Из забавного:

Факты из источника book\_3842 не совпали

Должно быть

Position: Президент

Hard: F | a | l | s | e

Where: Россия

Who: Путин Владимир

получилось:

Position: президент

Where: Россия

Who: Песков Дмитрий

Но так как большое количество несовпадение было связано с разной формой слова, то было принято решение проводить нормализацию при проверке. Пример ошибки:

Факты из источника book\_3744 не совпали

Должно быть

Position: главы

Hard: F | a | l | s | e

Where: МВФ

Who: Лагард Кристин

получилось:

Position: глава

Where: МВФ

Who: Лагарда Кристина

В связи с узкой специализированностью шаблонов, их требовательностью к нахождению ключевых слов и ручной отладкой по ходу разбора обучающих примеров, алгоритм демонстрирует высокие показатели точности. В этом можно убедится, проанализировав таблицу с оценками и типичные ошибки. Можно прийти к выводу, что парсер работает неплохо, но постоянные несовпадения в трактовке различных терминов, а также форм слов сильно портит конечный результат, а точнее оценку результата.

# Предложения по улучшению работы алгоритма.

Газеттир для тестового запуска был заполнен вручную с нуля. Адаптация существующего более полного газеттира должна сильно повысить полноту определения фактов, хотя в случае с должностями, имеющими сложную структуру, это может быть проблематично. В этом процессе очень помогут файлы ошибок, которые генерирует скрипт, в них все ошибки вскрываются достаточно просто, кроме того, для заполнения газеттира скрипт генерирует файл «NotInGaz.txt», в котором можно найти отсутствующие в газеттире сущности.

Созданные шаблоны, как было упомянуто в описании алгоритма, срабатывают только в тех случаях, когда описание должности и места работы осуществляется именной группой, непосредственно соседствующей с именем человека (поодиночке или в составе перечисления). Эти шаблоны покрывают значительную часть фактов Occupation, описываемых одним предложением, но существуют и другие, менее часто используемые способы выражения этого факта, при которых все его атрибуты присутствуют в одном предложении. В ходе выполнения домашнего задания они не рассматривались, так как был сделан акцент на правильном извлечении фактов, выраженных наиболее частотным способом, но шаблоны для других методов выражения факта могут быть добавлены без потери существующих.

Кроме того, приличная часть ошибок не является таковыми в действительности, потому что в банке ответов не предусмотрели дополнительные варианты, а наоборот намеренно заменили некоторые сущности на сокращения, чего нет в тексте, а также очень много ошибок с кавычками, окончаниям и т.п. Можно сделать предположение, что при дополнении газеттира и учете этих мелких нюансов. Качество выявления сущностей значительно возрастёт.