**Аннотация**

В данной работе решается задача извлечения метаинформации из документов с научным уклоном, а также реализация клиентского приложения, которое извлекает метаданные из текста и дает возможность опубликовать их в систему управления пространственной информацией – GeoNetwork.

**Введение**

Извлечение метаописаний – это задача автоматического извлечения структурированных данных из источников с неструктурированной или слабо структурированной информацией, которая связана с обработкой данных на естественных языках.

Извлечение метаописаний широко применяется в задачи информационного поиска. Эта процедура помогает более эффективно искать информацию, удовлетворяющую критериям поиска.

В общем случае задача состоит в том, чтобы из предложенного текста, выделить информацию об именованных объектах в тексте (информацию о классах, которые они соответствуют) и выделить ключевые слова. Набор классов фиксируется заранее.

Конкретно в моей задаче – нужно извлечь информацию из текста, предоставить интерфейс, где можно её отредактировать и дополнить, а также опубликовать в систему GeoNetwork.

Одной из подзадач в извлечении метаописаний – это выделение именованных сущностей (Named Entity Recognition, NER). Цель этой задачи выделить в тексте объектов определенного типа, это могут быть имена людей, географические объекты, биологические объекты и т.д.

Эта задача начала активно развиваться в середине 90-х, но и сейчас она остается актуальной, так как с каждым днем появляется потребность классификации, данный с новыми классами.

Под термином именованная сущность (named entity) понимается объект, который имеет такие характеристики как имя или идентификатор.

Сейчас существует много разных стратегий и подходов к решению поставленной задачи. В основном это системы, основанные на словарях, правилах или машинном обучении. Наиболее популярны методы машинного обучения с учителем. Такая система изначально тренируется на множестве примеров. Такую систему можно использовать на произвольных данных. Она имеет преимущество, по сравнению с моделями, основанными на словарях, так как может распознавать данные, которые раньше не встречала. Тем не менее, такая система тоже может ошибаться.

Выделение ключевых слов и словосочетаний

Ещё одна подзадача извлечения данных из текста. Ключевые слова – важные слова и словосочетания, которые дают высокоуровневое описание текста. Выделение таких слов, сделает информационный поиск более эффективным и точным.

Сейчас существуют алгоритмы, основанные на правилах и машинном обучении. Задача сводится к фильтрации текста и далее производится извлечение ключевых слов.

Связывание именованных сущностей с пространственным местоположением.

После извлечения географических объектов, хотелось бы получить информацию о местоположении объекта. Данную задачу можно решить с помощью словаря объектов и их координат и алгоритма неточного сравнения строк.

Эта подзадача не совсем связана с извлечением информации, но она так же актуальна, так как в системе GeoNetwork поиск может осуществляться по пространственным координатам.

Все перечисленное делает задачу извлечения метаописаний достаточно полезной и интересной не только с точки зрения исследователей, но и с точки зрения автоматизации процесса.

**Обзор существующих решений**

На просторах сети интернет существует множество систем, которые так или иначе решают задачу извлечения информации из неструктурированного или слабо структурированного текста.

Приведу некоторые примеры.

**Cremine** – это web сервис с открытым исходным кодом. У него неплохой интерфейс, но на предложенных ему документах он справился не самым лучшим образом.

Этот сервис использует машинное обучение, а именно CRF, SVM и K-means.

У него модульная структура, где для каждой подзадачи используются свои алгоритмы.

**GROBID** – сервис, который предлагается инсталлировать на свою машину и использовать.

Он имеет открытый исходный код и использует SVM для извлечения информации.

**PDF-Tools** – это платная десктопная программа для извлечения информации из PDF файлов.

**Text Razor** сервис для извлечения именованных сущностей из текста.

Несмотря на то, что представленные решения справляются с задачей извлечения информации, в той или иной степени, не одно из них не предоставляет данные в нужном формате, для публикации в GeoNetwork. Также в них нет специфических данных, таких как координаты.

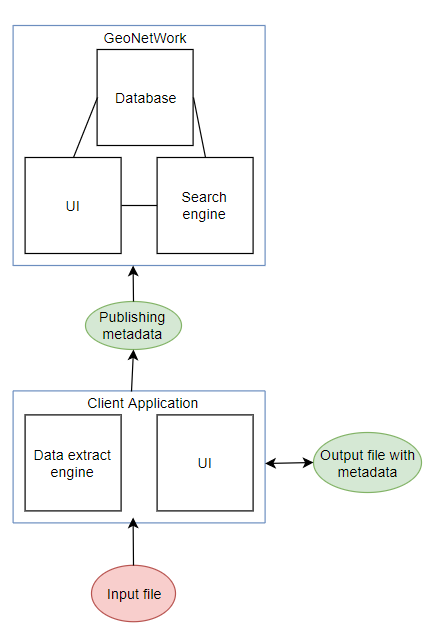
Следовательно, возникает необходимость разработать свою систему для извлечения метаописаний.

**Постановка задачи**

Задача данной работы состоит в разработке программного продукта для извлечения метаописаний из документов с последующим редактированием и публикацией в системе GeoNetwork. Для этого требуется:

1. Исследовать существующие методы извлечения информации из текстов;
2. Разработать методы извлечения информации из текстов;
3. Выполнить программную реализацию разработанных методов;
4. Реализовать клиентское приложение для системы GeoNetwork, в которое будет загружаться текстовый документ и извлекать метаданные из него. Также должна быть возможность отредактировать и дополнить эти данные, а затем опубликовать их.
5. Провести тестирование разработанной системы.

Примерная архитектура системы:



**Описание метода**

**Извлечение именованных сущностей**

Итак, мы имеем неструктурированный текст, и наша задача получить теги для слов из этого текста (I-PER, O, B-LOC и т. д.).

Пример:

Inna V. Stonik lives in Vladivostok.

I-PER I-PER I-PER O O B-LOC

В качестве обучающего корпуса был взят CoNLL2003, в котором имеются следующие виды именованных сущностей: места, имена людей, организации, разное.

Данные для обучения представлены в следующем виде:

Phil NNP I-NP I-PER

Simmons NNP I-NP I-PER

took VBD I-VP O

four CD I-NP O

for IN I-PP O

38 CD I-NP O

on IN I-PP O

Friday NNP I-NP O

as IN I-PP O

Leicestershire NNP I-NP I-ORG

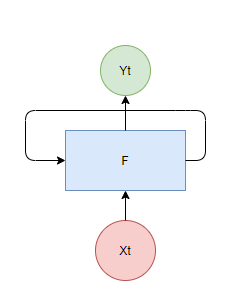
beat VBD I-VP O

Somerset NNP I-NP I-ORG

…

Проблема в задачи определения тега слова в том, что нельзя рассматривать каждое слово в отдельности. Например, Russian Federation – имеет тег ‘место’ и состоит из двух слов. Следовательно, нужно, чтобы модель имела память, и при встрече слова ‘Federation’ понимала, что ‘Russian’ тоже является местом. С этой задачей справляются рекуррентные нейронные сети.

Рекуррентные нейронные сети - сети, содержащие обратные связи и позволяющие сохранять информацию. Наличие этой связи дает возможность передавать информацию от одного шага сети другому.



Здесь сеть принимает и возвращает .

Конкретно в реализованной модели используется модификация рекуррентной нейронной сети – LSTM.

Долгая краткосрочная память или Long short-term memory (LSTM) – разновидность рекуррентных нейронных сетей, способная к обучению долговременным зависимостям.

Её особенность в том, что что она может хранить состояние. В ходе работы LSTM её состояние может меняется. Это позволяет принимать решение о том, что нужно запомнить, а что можно забыть.

Для обучения нейронной сети, представим слова из обучающего корпуса в виде векторов . Для этого используется алгоритм GloVe.

GloVe - это алгоритм машинного обучения для получения векторных представлений для слов. Обучение выполняется по совокупной статистике совпадения слов из корпуса, а полученные представления показывают линейные зависимости векторного пространства слов.

Так же мы будем учитывать и символы из которых состоит слово, так как это тоже даёт важную информацию. (Например, если слово начинается с заглавной буквы, то возможно оно является именем или локацией.) Это будет вектор , он описывает символьные характеристики слова.

Итак:

Каждый символ представлен в виде вектора .

После работы LSTM над , происходит конкатенация:

.

Далее происходит конкатенация :

– это и есть слово представленное в векторном виде.

Далее запускаем LSTM на полученном пространстве векторов и получаем новое пространство векторов .

Теперь, каждый вектор слова ассоциируется с вектором .

Чтобы сделать окончательное предсказание, выполним:

, где каждая компонента вектора – это вероятность в пользу того, что слово относится к определённому типу.

Теперь для окончательного предсказания нужно оценить вероятность последовательности слов.

Для этого используется алгоритм условно случайных полей (CRF).

Нужно оценить

– матрица перехода, – вектора оценок, которые фиксируют оценки первого или последнего заданного тега.

Матрица показывает линейную зависимость между соседними словами.

Таким образом происходит суммирование весов каждого слова и весов комбинации двух соседних слов.

Таким образом, вектора тегов между оцениваются так, чтобы была максимальна.

Так как сложность растет экспоненциально в зависимости от длины последовательности для решения этой задачи используется метод динамического программирования.

Пусть уже известно решение для для шагов от до , для последовательность начинается с для всех классов тегов.

Тогда решение для последовательности, начинающейся с будет иметь вид:

В итоге общая вероятность для последовательности тэгов определяется с помощью softmax так:

, где – нормирующий коэффициент:

- сумма оценок всевозможных последовательностей.

Для вычисления так же используется метод динамического программирования.

Пусть - сумма оценок всевозможных последовательностей, начинающихся с тэга на шаге t.

Тогда:

Для оценки потерь используется кросс-энтропийная функция

**Оценка качества модели**

При обучении алгоритма корпус разбивается на обучающую и тестовую выборку.

С помощью тестовой выборки можно оценить качество модели.

Оценка осуществляется с помощью следующих численных характеристик.

1. Точность:

Полнота присваивает всем классам одинаковый вес, что может быть не корректно в случае если распределение в обучающей выборке смещено в сторону каких-то классов.

Поэтому нельзя судить о качестве модели только по этой характеристике.

1. Точность:

Точность – это доля элементов, действительно принадлежащих данному классу относительно всех элементов, которые должны быть отнесены к этому классу.

1. Полнота:

Полнота – это доля найденных элементов, принадлежащих классу относительно всех документов этого класса в тестовой выборке.

 — истино-положительное решение;

 — истино-отрицательное решение;

 — ложно-положительное решение;

— ложно-отрицательное решение.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Истинные знач |  |
|  |  | Положительные | Отрицательные |
| Оценка сист. | Положительные | TP | FP |
|  | Отрицательные | FN | TN |

1. F-мера

Чем выше точность и полнота, тем лучше. Но в реальной жизни максимальная точность и полнота не достижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь метрику, которая объединяла бы в себе информацию о точности и полноте нашего алгоритма. Именно такой метрикой является F-мера.

F-мера представляет собой [гармоническое среднее](http://bazhenov.me/blog/2012/05/05/harmonic-mean.html) между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

**Извлечение библиографических ссылок**

Так как библиографическая ссылка имеет стандартизированный формат решено извлекать их с помощью регулярных выражений.

Каждую библиографическую ссылку можно разбить на составляющие:

список имен, заголовок, издание, год, количество страниц, серия.

Таким образом, можно составить регулярное выражение для каждой части, и их конкатенация будет давать полное регулярное выражение.

Полученные выражения:

name = "[A-Z]+[A-Za-z\\-]+,([\\s\*)([A-Za-z\\.]\*)](file:///\\s*)([A-Za-z\\.%5d*))"

names = "(" + self.regexName + "(\\s\*)[and,]\*(\\s\*)){1,}"

eds = "(\\(Ed\\.\\))|(\\(Eds\\.\\))"

names+eds = names + "(\\s\*)" + "("+eds+")\*"

year = "[0-9]\*\\.";

names+eds+year = names + "(\\s\*)" + "("+eds+")\*" + "(\\s\*)" +"("+year+")\*"

title+pages = "[A-Za-z0-9/\\.,\\s();\\-:]{1,300}[0-9]+(([\\-:][0-9]+\\.)|((\\s\*)pp\\.))"

reference = names+eds+year + "(\\s\*)" + title+pages

**Извлечение локаций с координатной привязкой**

Для извлечения локаций используется метод с рекуррентной нейронной сети, описанный ранее.

Далее эти локации отсеиваются с помощью словаря с названиями объектов и их координатами. Для сравнения кандидатов со словарём используется алгоритм нечёткого сравнения строк.

В алгоритме используется коэффициент Танимото.

Коэффициент Танимото:

*a* – количество символов в первом слове, *b* – количество слов во втором слове, *с* – совпадающие слова.

Только в данной реализации используется сравнение не символов, а N-грамм.

Такой подход помогает идентифицировать слова в разных формах и слова содержащие ошибки.

**Извлечение ключевых слов и словосочетаний**

# Для извлечения ключевых слов и словосочетаний используется алгоритм RAKE (Rapid Automatic Keyword Extraction).

В нём происходит генерация всевозможных словосочетаний. Подсчитывается частота этих словосочетаний в тексте. Далее рассчитывается граф частот соседних слов из которых состоят словосочетания. Далее для каждого слова рассчитывается его степень.

Степень – это сумма всех частот с которыми соседствует слово .

Чтобы вычислить окончательный приоритет словосочетания нужно для каждого слова вычислить его метрику .

Для улучшения качества используется нормализация слов, а так же фильтрация: стоп слова – словосочетания с этими словами игнорируется, опорные слова – предложения с этими словами используются для генерации новых словосочетаний.

**Ссылки**

1. VanderPlas J. Python data science handbook: Essential tools for working with data. – " O'Reilly Media, Inc.", 2016.
2. Пескова О. В. и др. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. – 2015.
3. Klinger R., Friedrich C. M. Feature subset selection in conditional random fields for named entity recognition //Proceedings of the International Conference RANLP-2009. – 2009. – С. 185-191.
4. Лабутин, И.А., Фирсов, А.Н., Чуприна, С.И. Распознавание именованных сущностей в текстах на естественном языке с использованием метода пробросо-цепочных условных случайных полей // Пермский государственный национальный исследовательский университет. – 8 с.
5. McCallum A., Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons //Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. – Association for Computational Linguistics, 2003. – С. 188-191.
6. Антонова А. Ю., Соловьев А. Н. Метод условных случайных полей в задачах обработки русскоязычных текстов //Информационные технологии и системы//Труды международной научной конференции. – 2013. – С. 1-6.
7. Zhu X. Cs838-1 advanced nlp: Conditional random fields. – Technical report, The University of Wisconsin Madison, 2007.
8. Колмогорцев С. В., Сараев П. В. Извлечение библиографии из текстов регулярными выражениями //Новые информационные технологии в автоматизированных системах. – 2017. – №. 20.