ВЕБ-СЕРВИС ДЛЯ АНАЛИЗА ТЕКСТОВ НА ОСНОВЕ АЛГОРИТМОВ НЕЙРОННЫХ СЕТЕЙ

NEURAL NETWORKS BASED WEB SERVICE FOR TEXT ANALYSIS

О.А. Павлюченко, О.В. Дубровина

V. Pauliuchenka, O. Doubrovina

Белорусский государственный университет
Минск, Беларусь
Belarus State University
Minsk, Belarus
E-mail: dubrovinaOVWtut. by

Технологии, основанные на обработке естественного языка, становятся все более распространенными. К примеру, телефоны и мобильные компьютеры поддерживают предсказывание текста и распознавание рукописи, движки веб-поиска дают доступ к информации, скрытой в неструктурированном тексте, машинный перевод позволяет получать текст написанный, например, на китайском языке и читать его на испанском [1].

Примерам подобных сервисов являются платформа для обработки текстовой и звуковой информации для различных тематических доменов corpus.by, национальный корпус русского языка ruscorpora.ru, программный продукт Яндекса SpeechKit Cloud API [2], который позволяет разработчикам приложений использовать речевые технологии, а также семантический тезаурус WordNet-Affect (http://wndomains.fbk.eu/wnaffect.html).

Проблемы при автоматизации обработки естественного языка часто заключаются в понимании самого языка, генерации формальных, машиночитаемых логических форм, восприятии диалога между человеком и машиной.

Разработанное веб-приложение включает в себя функции автоматического реферирования текстов, анализ и последующую визуализацию зависимостей слов в предложении, поиск сущностей в тексте, сравнение предложений по повторяемости и синонимическим конструкциям, сравнительную оценку семантического значения слов, анализ тональности предложения.

При разработке проекта в качестве основных программных средств были задействованы языки программирования Java, Python и их библиотеки, пакет прикладных программ Matlab, база данных PostgreSQL и Docker для обеспечения автоматизации развертывания и управления приложениями. В качестве дополнительных средств разработки использовались контейнер сервлетов Tomcat, Maven в качестве фреймворка автоматизации сборки проекта, система контроля версий git, среда разработки Intellij Idea, в качестве системы управления проектом - YouTrack, а также Amazon Web Services для развертывания приложения в облачном хранилище.

Программная реализация проекта основана на архитектуре микросервисов, что позволяет использовать различные языки программирования для создания модулей, интегрируемых в общий проект, и организации хранилища данных.

В основе проекта лежит реализованная программно LSTM-сеть - особый тип рекуррентной нейронной сети, способный обучаться долговременным зависимостям. Такие сети прекрасно справляются с решением многих задач и находят широкое применение в данное время [3]. Ключевой особенностью LSTM-сети является состояние ячейки, кроме того, такая сеть имеет возможность удалять и добавлять информацию в состояние ячейки. Для обучения реализованной нейронной сети был использован текст книги Л. Кэрролла «Алиса в стране чудес» [4].

Визуализация зависимостей между словами в предложении состоит из трех основных компонентов (рис. 1):

- слова и соответствующие теги частей речи, отображаемые на графике горизонтально;

- дуги разной длинны, соединяющие два слова с соответствующими метками, показывающие их тип отношения;

- стрелка в начале иди конце каждой дуги, указывающая на направление.

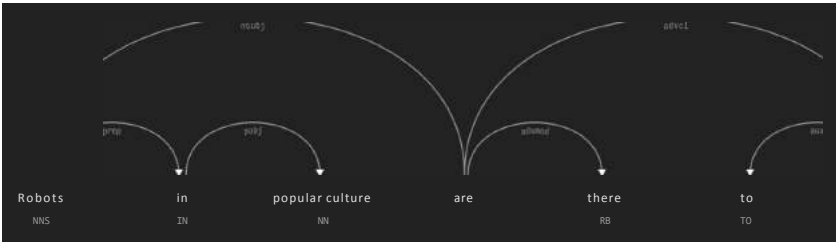

Рисунок 1 — Визуализация частей речи и связей в предложении

Дополнительно реализована возможность получить аналогичную древовидную схему (рис. 2).

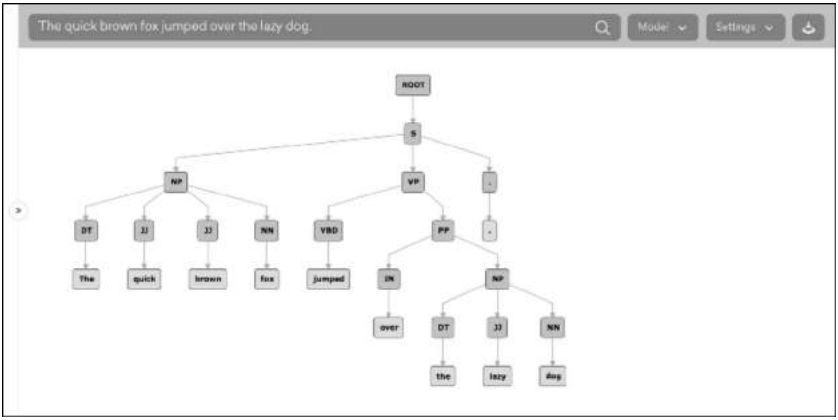

Рисунок 2 — Древовидная визуализация предложения

Распознавание сущностей включает в себя распознавание персонажа, организации, места действия, даты и так далее (рис. 3). Предложение вводится на форму, во вкладке Entity котрой выбираются сущности, которые мы хотим распознать, во вкладке Model выбирается язык (английский или немецкий).

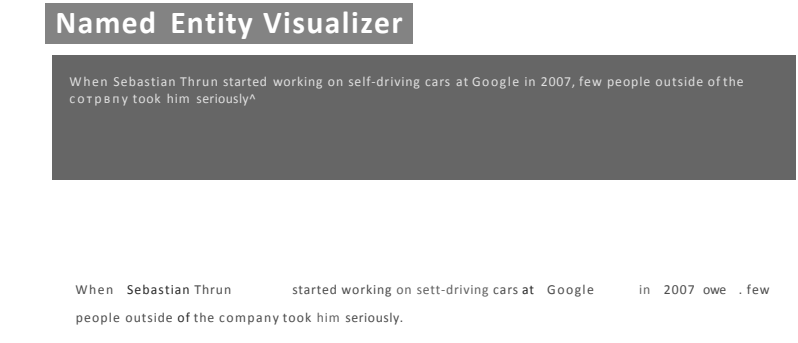

Рисунок 3 — Распознавание сущностей в тексте

Семантический анализ слов основан на добавлении тегов частей речи и названий сущностей объекта. Кроме того, необходимо

объединить именованные объекты и базовые существительные фразы в отдельные токены, чтобы получить один вектор.

Обработка семантики слов реализована на языке Python, для эффективной работы используется несколько потоков обработки.

В качестве примера рассмотрим анализ фразы *«All their good ideas»*. Первоначально циклически выделяются только существительные и прилагательные, то есть только «good ideas», далее словосочетание записывается в соотвествующий токен. Если токен состоит не из нескольких слов, то мы выходим из цикла и обрабатываем его заново как одиночный. Далее токены идут на вход в метод model.similarity, где вычисляются значения для сравнения с другими токенами при помощи латентно-семантического анализа [4].

Чтобы обработать словосочетание, необходимо поместить его на форму и выбрать во вкладке Sense настройки частей речи. По умолчанию, во вкладке Sense выбирается автоматически режим, однаконо можно настроить вывод только глаголов или существительных. Семантический анализ фраз в процентах показан на рисунке 4.

Рисунок 4 — Семантический анализ слов на основе сайта https://www.reddit.com/

Для сравнения предложений использовались несколько видов нейронных сетей: базовый алгоритм классификации, который вычисляет векторное среднее, нейронная сеть на основе данных сайта https://www.quora.com/ и нейронная сеть на основе данных сайта https://stackexchange.com/. Значительная разница в выходных данных может быть объяснена разными областями текста, на котором они были обучены.

Подобное сравнение используется для поиска спама, а также актуально для большинства дискуссионных форумов, где одни и те же вопросы задаются повторно.

С помощью метода наивного байесовского классификатора [6] был построен классификатор текстов, который используется для базового сравнения предложений.

Чтобы сравнить предложения необходимо добавить их в форму (рис. 5), которая отображает результат вычислений приведенными выше методами.

Рисунок 5 — Сравнение предложений

Чтобы построить дерево тональности, нужно добавить предложение на форму и во вкладке Model выбрать язык (английский или немецкий). Деревро определения тональности показано на рисунке 6. При добавоении сразу нескольких предложений, то их графические представления будут отображаться во вкладках друг под другом.

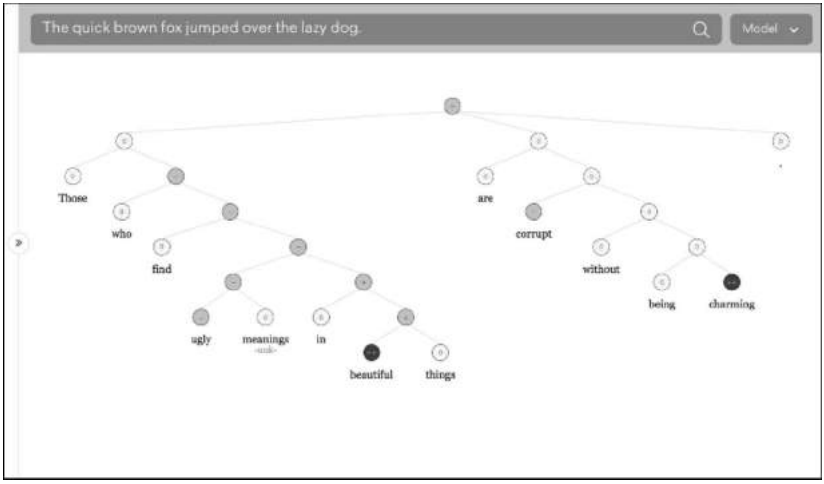

Рисунок 6 — Дерево определения тональности текста

На основе LSTM-сети реализован модуль упрощения текста, результатом которого является краткий реферат (см. рис. 7). Существует возможность выбрать язык в выпадающем меню «Model» и количество предложений в выпадающем меню «Number of sentences», до которого необходимо упростить текст.

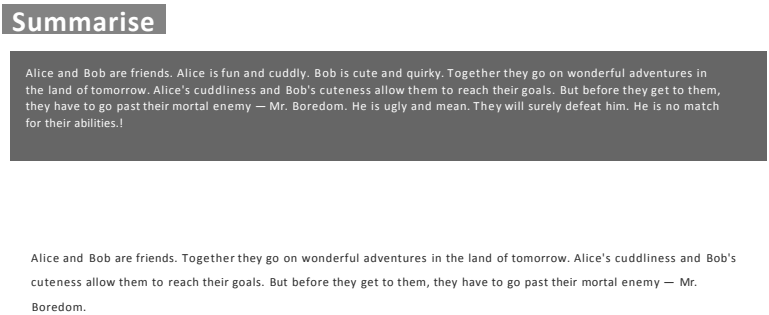

Рисунок 7 — Упрощение текста

Разработанный программный продукт включает в себя функции автоматического реферирования текстов, анализ и последующую визуализацию зависимостей слов в предложении, поиск сущностей в тексте, сравнение предложений по повторяемости и синонимическим конструкциям, сравнительная оценка семантического значения слов, анализ тональности предложения.

Веб-приложение представляет собой законченный программный продукт и может использоваться для исследований в области естественного языка и прикладной лингвистики, анализа структуры и семантики языка и анализа больших текстовых данных.

Проект разворачивается на платформе облачных технологий Amazon Web Services, его модернизация может быть направлена организацию специфики такой работы, а также внедрение в него новых программных модулей.

ЛИТЕРАТУРА

1. Лабараторыя распазнавання і сінтэзу маўлення [Электронный ресурс]. Режим доступа: http://ssrlab.by - Дата доступа: 18.01.2017.

2. Resources for Text, Speech and Language Processing [Электронный ресурс]. Режим доступа: http://www.cs.technion.ac.il/ ~gabr/resources/pointers.html - Дата доступа: 20.01.2017.

3. Investigations on dynamic neural networks [Электронный ресурс]. Режим доступа: http://people.idsia.ch/~juergen/ SeppHochreiter1991 Thesis AdvisorSchmidhuber. pdf - Дата доступа: 03.02.2017

4. Кэрролл, Л. Приключения Алисы в стране чудес /Л. Кэрролл. - М.: 2012. - 168 с.
5. Latent Semantic Analysis [Электронный ресурс]. Режим доступа: http://lsa.colorado.edu - Дата доступа: 11.03.2017
6. Bayes classifier [Электронный ресурс]. Режим доступа: http://dataaspirant.com/naive-bayes-classifier-machine-learning/ - Дата доступа: 12.05.2017.