РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРЕОБРАЗОВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ В ВИЗУАЛЬНУЮ ФОРМУ

К. С Мулярчик, В. П. Можейко

Белорусский государственный университет, Минск, Беларусь
E-mail: k.mulyarchik@gmail.com, vladislav.mozheiko321@gmail.com

В данной статье предложена классификация визуальных форм по их базовым элементам, благодаря которой можно автоматизировать их построение. Продемонстрирован пример работы алгоритма построения визуальных форм, разработанный на основе приведенной классификации. Сделаны выводы о возможных реализациях и применениях данной разработки.

Ключевые слова: *визуальная форма, обработка естественного языка, spacy.*

В современном мире количество информации, которую приходится ежедневно обрабатывать человеку, достигает огромных объемов. Для упрощения её восприятия мы часто используем ее визуальное представление [1]. Рисунки и изображения помогают нам лучше понять общую картину происходящего.

Данная работа посвящена рассмотрению подхода к автоматизации построения визуальных форм по заданному тексту. Необходимо составить классификацию базовых визуальных элементов, получение которых возможно с помощью простых алгоритмов, разработать и реализовать алгоритм построения сложных визуальных форм на основе анализа текста.

Для составления классификации базовых элементов был проведен анализ различных диаграмм, графиков, гистограмм и других схематических изображений. В ходе анализа рассматривались изображения, которые мы используем на картах (Google Maps, Яндекс карты [2]), диаграммы, которые можно получить с помощью алгоритмов Matlab и Python. Также проводился анализ изображений, используемых в различных презентациях и статьях [4, 5] и анализ инфографиков, доступных в интернете [3, 6]. В результате было выделено несколько базовых элементов визуальных форм: 1) Набор объектов. Это любая коллекция объектов, принадлежащих к какому-либо классу на основании общего признака. Для примера, на рис.1 присутствует класс – космические тела. Каждый отдельно взятый объект имеет разные характеристики, но все они принадлежат к одному классу; 2) Сети. Визуальное отображение взаимосвязей между объектами. Может быть представлено в виде древа, графа или другой структуры, отражающей взаимосвязи между набором объектов. Под эту категорию могут попасть так же и визуальные формы, которые изначально не выглядят как сеть. Внешний вид зависит от метода визуа-

лизации информации, как можно увидеть на рисунке 2. Два типа визуальных форм отражают одну и ту же информацию.

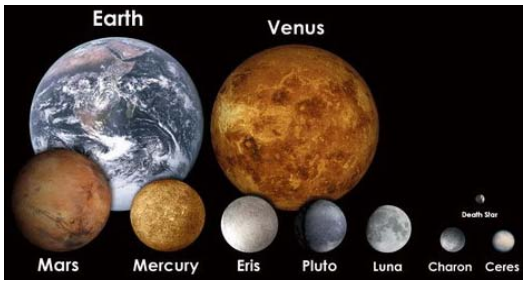

Рис. 1. Визуальное сравнение размеров космических тел

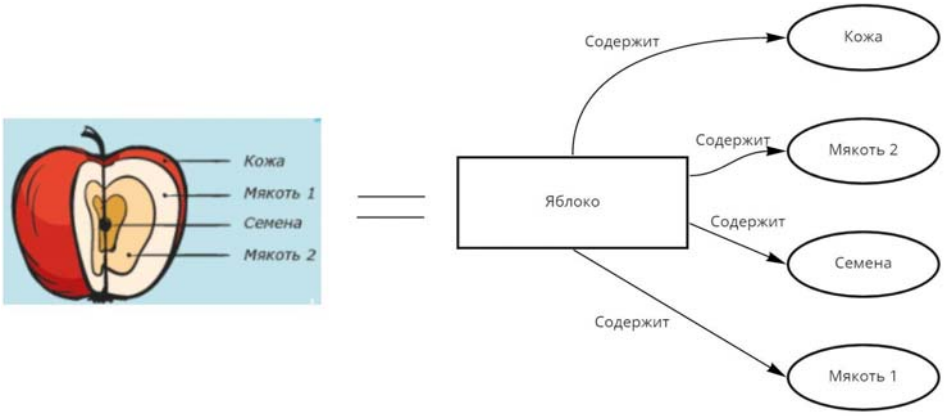

Рис. 2. Яблоко в разрезе как визуальная форма «сеть»

1. Визуальные сравнения. Выражаются в виде различных характеристик набора объектов одного и того же класса. Как показано на рисунке 1, с помощью визуального сравнения представлены две характеристики: размер и различные материалы поверхности планет.

2. Карта. Представляет любое упорядоченное расположение элементов на визуальной форме. Может быть выражена, например, сортировкой по размеру, как на рисунке 1, или же заданием точного расположения объектов (точек на графике), как на рисунке 3.

3. Изображение объекта. Внешний вид объекта может использоваться не только для визуального сравнения набора объектов, но и для уточнения характеристик отдельно взятого объекта.

С использованием данной классификации для определения базовых этапов при реализации программы, был разработан алгоритм для отражения взаимосвязей между объектами. В первую очередь производится предварительная обработка текста в виде разбиения его на токены и получения двумерных массивов, где каждая строка – предложение, а каждый элемент – слово. Далее производится базовая очистка текста от ненужной информации: производится лемматизация, удаление отдельных стоп-слов и символов, таких как “(”, “[”, “/”, “.” и т.д.

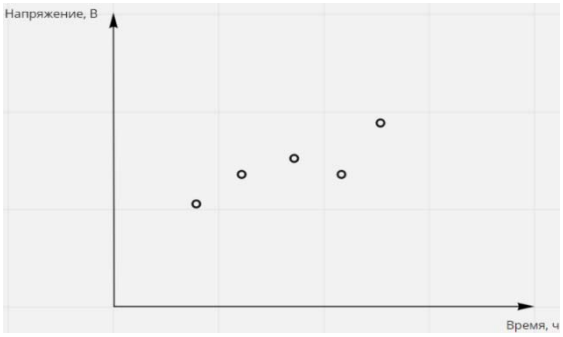

Рис. 3. Элемент карты на графике

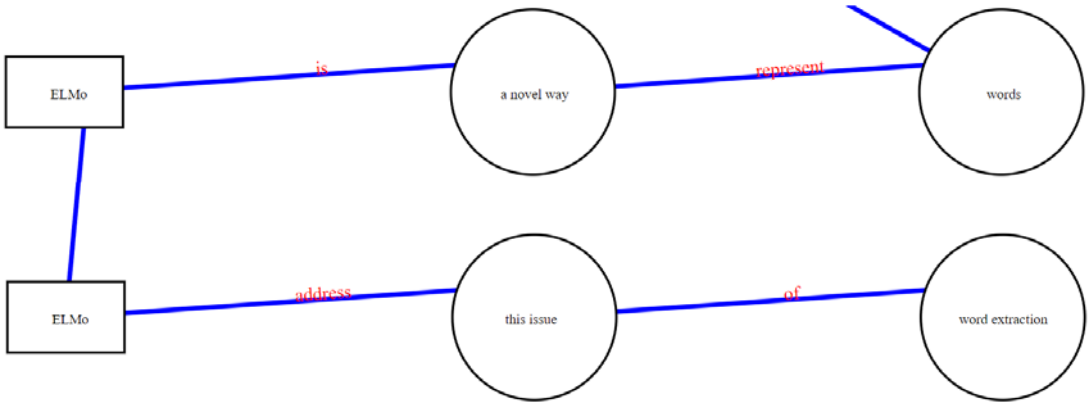

Рис. 4. Внешний вид визуальной формы, преобразованной из текста

Существительные с описанием – основной класс, который представляет набор объектов в данном алгоритме. Для определения элементов данного класса, используя нейронную сеть en_core_web_lg из библиотеки spacy определяем части речи и строим синтаксическое древо. Синтаксическое древо – ориентированное древо, с узлами-токенами и ветвями, являющимися различными типами взаимосвязей (глагольные, именные, предложные и т. д.). Определяя все существительные по части речи, объединяем их с соседними токенами, имеющими именные связи с существительным в синтаксическом древе. В результате получаем набор объектов с их описанием.

На следующем этапе, для подчеркивания наиболее важных объектов – сущностей (имя человека, название организации и т. д.), используем визуальное сравнение объектов в наборе элементов. Применяя готовую нейронную сеть, проводим анализ каждого токена, определяем именованные сущности и записываем эту характеристику объекта в Boolean-значение для дальнейшей визуализации. На визуальной форме значение данного параметра отображается в виде отличий в форме рамки вокруг объекта. Прямоугольная для именованной сущности и круглая для обычного объекта.

Следующий этап – получение взаимосвязей между набором объектов. Для этого реализуется элемент «сеть» на основании графа, элементами

которого являются токены. Каждое ребро отражает наличие одного из типов связи между словами в предложении. Количество ребер – расстояние от одного токена до другого. Граф получается с помощью преобразования синтаксического древа, убирая ориентированность. Используя поиск кратчайшего пути попарно между всеми объектами с условием, что между двумя объектами не может находиться другого объекта, определяем все пути между объектами. Каждый путь объединяем в одну строку, добавляя маркеры элементов, которые она связывает. Строки на визуальной форме отображаем в виде синих ребер между объектами из набора элементов с текстом описания взаимосвязи.

В данной визуальной форме также присутствует базовый элемент «карта», хотя отдельная модификации для этого не проводились. Все токены (слова) при обработке располагаются в том порядке, в котором они появляются в тексте. Соответственно, на визуальной форме они также упорядочены по тому же принципу.

Описанный выше алгоритм был реализован программно в среде разработки Visual Studio Code на языке программирования Python. Для реализации разбиения текста на токены и определения частей речи и взаимосвязей были использованы библиотеки nltk и spacy, для работы с деревьями и графами – библиотека networkx. Результат работы программного обеспечения представлен на рисунке 4.

В заключении стоит отметить, что данное программное обеспечение является базовым примером преобразования текстовой информации в визуальную при использовании классификации визуальных форм. В зависимости от целей, которые перед собой ставит разработчик, можно производить построение форм, опираясь на различные комбинации основных элементов визуальных форм.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Визуальное мышление. Как «продавать» свои идеи при помощи визуальных образов / Д. Роэм; пер с англ. О. Медведь. М.: Манн, Иванов, Фербер, 2013. 300 с.
2. Яндекс Карты. [Электронный ресурс]. – Режим доступа: https://yandex.by/maps/. – Дата доступа: 01.01.2022.
3. Графический дизайн. Инфографика. [Электронный ресурс]. – Режим доступа: https://freelance.ru/KuznecovEvgenii#info-grafika. – Дата доступа: 01.01.2022.
4. Графические формы свертывания информации. Визуализация информации при создании инфографики [Электронный ресурс]. – Режим доступа: https://en.ppt-online.org/51360. – Дата доступа: 19.01.2022.
5. Reveal The Data blog. [Электронный ресурс]. – Режим доступа: https://revealthedata.com/blog/all/chto-takoe-vizualizaciya-dannyh-kakaya-ona-byvaet-i-ne-byvaet/. – Дата доступа: 19.01.2022.
6. Information Is Beautiful blog. [Электронный ресурс]. – Режим доступа: https://informationisbeautiful.net/blog/. – Дата доступа: 20.01.2022.