Санкт–Петербургский государственный университет

Беляков Юрий Дмитриевич

Выпускная квалификационная работа

Разработка сервиса по автоматическому созданию видео презентации из текста

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Математическое и программное обеспечение вычислительных машин»

Научный руководитель:
кандидат технических наук,
кафедра технологии программирования
Блеканов Иван Станиславович

Научный соруководитель:
ассистент,
кафедра технологии программирования
Шиманская Галина Станиславовна

Рецензент:
кандидат физико-математических наук,
кафедра компьютерных технологий и систем
Погожев Сергей Владимирович

Санкт-Петербург
2021 г.

Содержание

Введение

Современное общество невозможно представить без интернета. Он играет важную роль в жизни огромного количества людей. При помощи интернета мы можем общаться, рассказывать о своей жизни, получать необходимую информацию и т.д. Можно обобщить различные активности, ради которых используется интернет: мы можем потреблять контент (информационное содержание) или создавать его. Такой формат взаимодействий и огромное число пользователей создают благоприятные условия для рекламы. Интернет - идеальное место для продвижения своего продукта. Бизнес любой величины сможет найти в социальных сетях свою целевую аудиторию и получить прирост покупателей и, соответственно, прибыли, а потребители контента - найти нужные им продукты или услуги. С точки зрения маркетинга, помимо очевидного преимущества в объемах охватываемой аудитории, в сравнении с прочими источниками информации, интернет обладает другой, не менее важной, особенностью - формат контента. Мы не можем разместить видео в газете или на листовке; не можем поделиться своим блогом или статьей на телевидении. Для рекламы в интернете может использоваться любой тип контента. Однако, в особенности, популярностью пользуется видеоконтент. В связи с этим, владельцы малых бизнесов и маркетологи вынуждены создавать видеоконтент для продвижения своих продуктов, чтобы упростить восприятие информации и охватить как можно больше аудитории. Но нельзя сказать, что создать видео - легко. Это трудо- и времязатратный процесс. Цель данной работы - упростить создание конкретного вида видео-контена, а именно - видеопрезентаций.

Видеопрезентация - последовательность из слайдов (Рис. 1), каждый из которых состоит из фонового изображения/видео и небольшого фрагмента текста, вместе с музыкальным сопровождением. Такой формат контента может использоваться для удобной и краткой презентации чего-либо.

Принцип работы сервиса состоит в следующем. На вход подается некоторый текстовый документ. Это может быть заранее написанный сценарий для видео; запись из текстового блога; новостная статья и т.д. Разработанный сервис:

1. Сокращает объем текстовой информации до необходимого минимума, при котором сохраняется главная мысль и тема текстового документа, но отсутствует вся излишняя информация. (Краткое содержание)

2. Полученное краткое содержание разбивается на слайды

3. Каждому из слайдов ставится в соответствие изображение/видео, которое подходит по смыслу как к общей тематике текста, так и к содержанию фрагмента на слайде.

4. К презентации подбирается жанр/настроение/тематика музыкального сопровождения, подходящее к тексту.

С помощью уже существующих инструментов для обработки видео результаты работы предложенного сервиса можно объединить и получить готовый видеофайл.

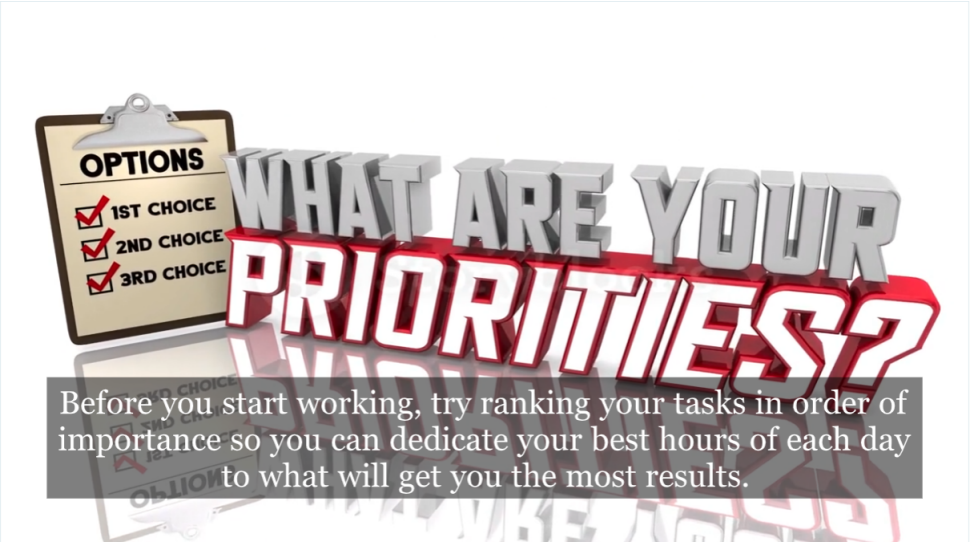

Рис. 1: Пример слайда.

Постановка задачи

Цель данной работы - создание веб-сервиса по подбору содержания видеопрезентаций (текстовая информация, медиафайлы и музыкальное сопровождение) на основании текстового документа, используя существующие методы машинного и глубокого обучения. Один из главных критериев качества сервиса - поддержка как можно большего количества языков. Для достижения этой цели были выделены следующие задачи:

1. Разбить общую задачу составления видеопрезентации на существующие и хорошо изученные подзадачи обработки естественного языка.
2. Провести обзор предметной области.
3. Провести анализ существующих методов и моделей решения выделенных подзадач и выбрать те модели и методы, которые имеют устойчивость к смене домена и к смене языка.
4. Реализовать API сервис, использующий современные модели и методы обработки естественного языка.

Глава 1. Обзор предметной области

1.1 Автоматическая суммаризация текста

Обработка естественного языка (Natural Language Processing, NLP) - общее направление искусственного интеллекта и математической лингвистики. В нем изучаются проблемы компьютерного анализа и синтеза естественных языков. Задачи NLP направлены на создание таких систем, которые способны выполнять задачи, связанные с языком, на таком же уровне как и человек. Одна из таких задач - автоматическая суммаризация текста (Automatic Text Summarization, ATS).

Краткое содержание любого документа имеет невероятную ценность. Оно сохраняет нам время, позволяет оценить релевантность текста к предмету нашего поиска, с его помощью мы способны презентовать любую текстовую информацию не вдаваясь в подробности, и др. Создание краткого содержания вручную - это трудоемкий и времязатратный процесс, требующий знания полного содержания документа и способности выделить ключевые аспекты. Изложения могут разниться от одного человека к другому. С ростом объема информации необходимость исследований в области только увеличивалась. Мы можем получить нужные нам знания как угодно и где угодно, однако избыточные данные могут затруднять и замедлять нам поиск. Таким образом, изучение автоматических способов отделения полезной информации от всего остального имеет очевидные преимущества.

Автоматическое создание таких ценных отрывков текста - давно известная проблема в области обработки естественного языка. Работа Ханса Питера Луна [1], посвященная решению этой задаче, была одной из первых. С тех пор появилось огромное количество различных методов решения этой задачи.

Различают 2 вида суммаризации:

- Экстрактивный подход - составление краткого содержания как подмножество предложений исходного текста, которые содержат наибольшее количество информации.

- Абстрактивный подход - составление нового текстового документа,

содержательно обобщающего исходный.

У каждого из подходов есть свои преимущества и недостатки. Методы абстрактивной суммаризации решают seq2seq задачу генерации текста. За счет этого, при идеальной работе, будет получен связный текст меньшего объема, похожий на составленный человеком. Однако существующие методы очень чувствительны к домену текста, а за счет того, что текст генерируется с нуля, малейшие ошибки могут вести к генерации нечитабельного текста. Плюс к этому, современные методы основаны на моделях глубокого обучения, таких как Bert [2] и T5 [3], следовательно, они имеют относительно долгое время работы.

Методы экстрактивной суммаризации выбирают лучшие (в каком-то понимании) предложения из исходного текста и составляют из них краткое содержание. При таком подходе не может идти речь о какой-либо оригинальности или похожести на составленный человеком текст. Однако такие методы устойчивы к ошибкам, к смене домена и быстро работают, что является большим плюсом при необходимости составления готового краткого содержания без последующего редактирования. В связи с этим, в данной работе используются методы экстрактивной суммаризации и, в дальнейшем, под суммаризацией будет пониматься именно экстрактивный подход.

На сегодняшний день лучшие методы суммаризации основаны на использовании языковой модели BERT. Однако такие методы, как правило, хорошо решают задачу лишь для одного языка - английского. Это связано с тем, что существующие крупные наборы данных, которые необходимы для обучения моделей, - на английском языке. Одной из поставленных задач является максимально возможная независимость модели от языка. В связи с этим, для решения задачи суммаризации будет использоваться метод без учителя, который не привязан к конкретному языку.

1.2 Извлечение ключевых слов

После составления краткого содержания необходимо подобрать изображения/видео, которые наилучшим образом подходят к тексту, а затем разбить набор предложений и медиафайлов на слайды. Для подбора медиафайлов были использованы существующие сервисы (Pixabay, Shutterstock, Storyblocks и т.д.), с помощью которых можно получить доступ к огромному количеству стоковых изображений и видео. Принцип работы таких сервисов следующий: по некоторому текстовому запросу (как правило, набор ключевых слов) можно получить набор медиафайлов, причем у каждого полученного файла есть некоторое текстовое описание того, что изображено на изображении или происходит на видео. Таким образом, для составления запроса к такому сервису и, впоследствии, подбора подходящих к тексту изображений необходимо выделить в тексте ключевые слова. Такая задача известна как "Извлечение ключевых слов" (Keyword extraction). Существуют методы извлечения ключевых слов с учителем (supervised) и без учителя (unsupervised). В данной работе было принято использовать unsupervised методы, потому что такие алгоритмы не зависят от обучающей выборки, а следовательно устойчивы к смене домена/языка, что является очевидным преимуществом в рамках данной работы. Можно обобщить принцип работы unsupervised алгоритмов по этапам их работы. Как правило, принцип работы таких методов выглядит следующим образом:

1. Выбор кандидатов. Основываясь на различных эвристических методах выбираются те лексические единицы (слова, фразы), которые, предположительно, могут выполнять роль ключевых слов. Такими методами могут быть: проверка на принадлежность слова к списку стоп-слов; выбор слов, относящихся к определенным частям речи; и т.д.

2. Оценка кандидатов. По какому-либо методу каждому из кандидатов ставится в соответствие некоторая оценка.

3. Составляется список ключевых слов. Первые N кандидатов с наилучшими оценками становятся ключевыми.

В [4] было проведено сравнение (Таблица 2) популярных unsupervised методов. В таблице 2 обозначения "@10"и "@20"означают, что методы извлекали 10 и 20 ключевых слов, соответственно. Сравнение проводилось на различных наборах данных:

1. Krapivin [5].
2. Semeval [6].
3. NUS [7].
4. Inspec [8].
5. 500N-KPCrowd [9].

Сравнивались 9 различных методов относящихся к двум типам алгоритмов: статистические и графовые. Наборы данных различались по языку, домену и объему текста. По результатам сравнения было выявлено, что в большинстве случаев статистические методы показывали себя лучше, чем графовые. Также, в ходе сравнения было выявлено, что статистические методы работают значительно лучше на объемных текстовых документах. Особенно, среди статистических методов выделялся KPMiner [10] (KPM).

	Semeval		NUS		Krapivin		Inspec		500N-KPCrowd	
F_1	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
YAKE	0.160	0.169	0.188	0.180	0.124	0.109	0.197	0.212	0.107	0.168
Tfldf	0.154	0.176	0.201	0.205	0.126	0.113	0.197	0.212	0.179	**0.243**
Tfldf2	0.172	0.191	0.229	0.217	0.156	0.138	0.184	0.193	**0.180**	0.233
KPM	**0.208**	**0.219**	**0.259**	**0.243**	**0.190**	**0.161**	0.107	0.106	0.143	0.178
RAKE	0.114	0.147	0.134	0.142	0.091	0.096	0.216	0.233	0.064	0.066
MR	0.146	0.161	0.147	0.149	0.112	0.100	0.245	0.269	0.156	0.224
TR	0.134	0.142	0.126	0.118	0.099	0.086	0.235	0.249	0.148	0.209
PR	0.131	0.127	0.146	0.128	0.102	0.085	0.253	0.273	0.145	0.206
SR	0.036	0.053	0.044	0.063	0.026	0.036	**0.278**	**0.295**	0.096	0.164
RVA	0.096	0.125	0.096	0.115	0.093	0.099	-	-	-	-

Рис. 2: Результаты сравнения Unsupervised методов извлечения ключевых слов. Таблица взята из [4].

1.3 Схожесть текстовых документов

После извлечения ключевых слов из текста и получения набора изображений подходящих по тематике к тексту необходимо подобрать каждому предложению из составленного краткого содержания фоновое изображение/видео. Для решения этой задачи можно воспользоваться текстовыми описаниями элементов набора медиафайлов и свести задачу подбора подходящих изображений к задаче подбор подходящих текстовых документов следующим образом:

1. Составить векторное представление предложения из сокращенного ранее текста (главный вектор).

2. Составить векторные представления текстовых описаний полученных ранее медиафайлов (векторы-кандидаты).

3. С помощью меры схожести, а именно косинусной схожести, найти ближайший к главному вектору вектор-кандидат. Косинусная схожесть определяется следующим образом:

$$cos(\theta) = \frac{v \cdot w}{\|v\|\|w\|} = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}}$$

   где v и w - векторные представления текстовых документов между которыми требуется найти схожесть.

Это задача нахождения схожести текста (Text Similarity). Решение такой задачи сводится лишь к нахождению лучшего способа представить текст в виде вектора. Это можно сделать с помощью, например, таких методов, как:

- TF-IDF. Метод векторизации текста основанный на одноименной статистической мере. Мера TF-IDF вычисляется по следующей формуле:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D),$$

  где $tf(t, d)$ - частота встречи слова t в документе d (под документом, в контексте нашей задачи, понимается предложение), а $idf(t, D)$ -

обратная частота встречи слова в коллекции документов D:

$$idf(t, D) = log\frac{\mid D \mid}{1 + \mid \{d \in D : t \in d\} \mid}$$

Преимущества метода: независим от языка и домена.

Недостатки: при векторизации не учитываются семантические связи между словами (текст рассматривается как набор слов).

- Doc2Vec [11]. Метод векторизации документа, основанный на методе Word2Vec [12]. Word2vec - это инструмент (набор алгоритмов) для расчета векторных представлений слов, реализует две основные архитектуры — Continuous Bag of Words (CBOW) и Skip-gram. На вход подается корпус текста, а на выходе получается набор векторов слов. Doc2Vec создает веторное представление документа фиксированной размерности вне зависимости от его длины.

  Преимущества: учитываются семантические связи между словами.

  Недостатки: небольшое количество открытых предобученных моделей; сильная зависимость от языка и домена.

- SentenceBERT [13]. Метод векторизации предложений, основанный на языковой модели BERT. С помощью сиамской архитектуры (Рис. 3) оптимизируется операция усреднения веторных представлений токенов на задаче нахождения текстового сходства. Операция усреднения векторных представлений оптимизируется следующим образом:

  1. Две модели архитектуры BERT с идентичными весами составляют векторные представления токенов.
  2. Векторные представления токенов усредняются.
  3. Между полученными векторными представлениями предложений ищется схожесть.
  4. Значение полученной схожести сравнивается с правильным ответом.

5. Обновляются веса модели.

Преимущества: учитываются семантические связи между словами; использование State-of-the-Art языковой модели BERT; большое количество доступных предобученных моделей для различных доменов и языков.

Недостатки: отсутствие хороших предобученных моделей, поддерживающих большое количество языков.

- Universal Sentence Encoder [14]. Один из лучших существующих способов векторизации предложений. Предобученная мультиязыковая модель USE переводит любой входной текст в векторное пространство размерности 512. USE использует 6 слоев трансформер-блоков (Рис. 4) для составления векторных представлений токенов, а затем применяет поэлементное сложение и нормализует вектор для получения вектора фиксированной длины (512). Такая архитектура обучалась сразу на нескольких задачах NLP, а именно: Modified Skip-thought, Conversational Input-Response Prediction и Natural Language Inference.

  Преимущества: Предобученная модель поддерживает большое количество языков; учитываются семантические связи между словами; в отличие от предыдущих моделей, обучалась изначально с целью составления векторного представления предложений (другие методы адаптируют ранее разработанные методы векторизации токенов/слов) Минусы: Нет полной независимости от языка (обучение происходило на 16 языках)

Существуют и другие модели, в которых используется идея оптимизации операции усреднения для использования составленных ими векторных представлений для нахождения схожести текстов. Но, как правило, такие модели обучены только на одном языке. USE, в свою очередь, поддерживает 16 языков. Если необходимо сохранить полную независимость от языка, то следует использовать статистические методы, например, TF-IDF.

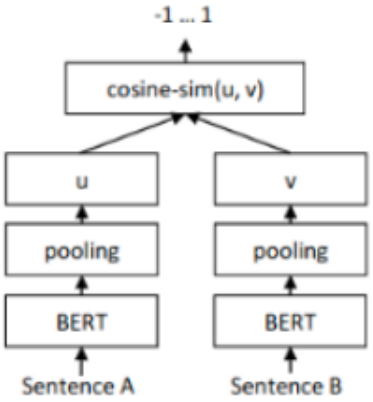

Рис. 3: Архитектура SentenceBERT для задачи нахождения сходства предложений.

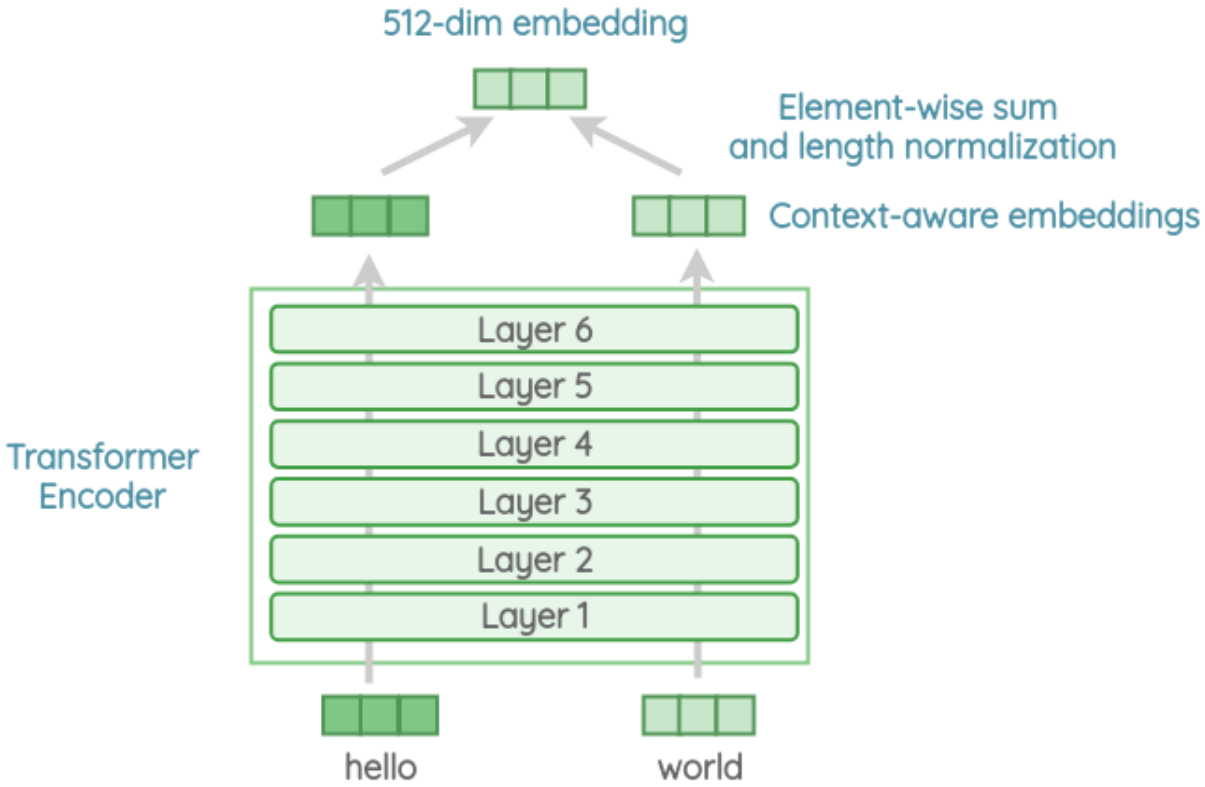

Рис. 4: Архитектура Universal Sentence Encoder.

1.4 Классификация без подготовки

Последний этап - выбор музыки. Цель сервиса - определить жанр / настроение музыкального сопровождения так, чтобы оно подходило к тексту. Для такой задачи совсем необязательно работать с аудио-файлами. Достаточно поставить задачу классификации текста. Однако при определении жанра музыки не хотелось бы иметь заранее определенный набор классов. Музыка может различаться не только по жанру (поп, рок, хип-хоп), но и по настроению (вдохновляющая, грустная, веселая), по тематике (детская, бизнес, новости) и т.п. Поэтому, для того, чтобы у пользователя была возможность выбора характеристики музыки, по которой нужно выполнить классификацию, в работе используется метод классификации без подготовки (Zero-shot Classification). Zero-shot классификация текста - задача классификации, в которой модели могут классифицировать текст, не обучаясь при этом на наборе данных, созданном для этой задачи классификации. Если для решения задачи классификации модели нужен набор данных для обучения с заранее определенным конечным числом классов, то для zero-shot такой набор не нужен - модель способна предсказать к какому из предложенных классов вероятнее всего относится текст и без такого набора.

Такое возможно благодаря задаче NLI (Natural Language Inference) - есть некоторое высказывание (например - "Сейчас идет футбольная игра между несколькими мужчинами"), некоторая гипотеза (например - "Мужчины сейчас играют в футбол") и необходимо определить один из трех исходов:

- *Entailment* - логическое следствие (из высказывания следует гипотеза)
- *Contradiction* - противоречие (из высказывания не следует гипотеза)
- *Neutral* - нейтральность (гипотеза и высказывание не связаны)

В данном случае правильный ответ - Entailment. Такой подход к решению задачи классификации без определенных заранее классов с помощью

предобученных языковых моделей и успехов в задаче NLI был предложен в [15]. С помощью предобученных мультиязычных моделей, решающих задачу NLI, можно переформулировать задачу опредения класса музыки следующим образом: Пусть есть множество Y, состоящее из n классов, заданных текстовой строкой (например, {*детская, бизнес, новости, наука*}), к которым может относиться текстовый документ d, тогда, с помощью модели NLI, определяющей вероятность $P(h|p)$ того, что пара *высказывание (p) - гипотеза (h)* относится к классу *Entailment*, то для классификации достаточно найти:

$$argmax_{y \in Y} \ P(y|d)$$

Тогда класс y, для которого будет достигаться максимум вероятности, можно считать классом, к которому относится текстовый документ.

Глава 2. Программная реализация сервиса с использованием предобученных моделей

Примеры работы отдельных составляющих сервиса будут продемонстрированны на статье "The Only Way to Be Truly Confident in Yourself" на Medium.com [16].

2.1 Архитектура решения и используемые технологии

Описанное решение реализовано в виде API-сервиса на языке программирования Python с использованием фреймворка Flask. Для работы с языковыми моделями используются библиотеки Transformers (HuggingFace) и Tensorflow. Для предобработки текста используется модуль NLTK.

Сервисом можно воспользоваться отправив GET-запрос с указанием двух параметров: url - ссылка на статью, видеопрезентацию которой необходимо составить, и lang - язык статьи. На выходе - содержимое видеопрезентации в формате JSON (Рис 5). Ответ состоит из поля *theme*, в котором указана тематика музыкального сопровождения, *slides* - список слайдов, каждый из которых состоит из текста *text* и *pics*.

```
{
    "music_theme": "motivational"
    "slides" : [
        {
            "text": "Текст слайда",
            "pics": imageObject
        }
        ...
    ]
}
```

Рис. 5: Формат ответа API-сервиса.

2.2 Используемые в сервисе методы и модели

2.2.1 LexRank

Для определения наилучшего Unsupervised метода суммаризации было проведено сравнение 6 методов:

1. TextRank [17]. Графовый алгоритм, главной идеей которого является "рекомендация". Каждое предложение выполняет роль вершины направленного графа. Если вершина A соединяется с вершиной B, это значит что B получила "голос" или "рекомендацию" от A. Чем больше число голосов у вершины, тем больше важность предложения. Более того, важность вершины, которая отправляет голос, влияет на важность голоса.

   Рекомендацию можно определить по-разному. Для задачи суммаризации такой "голос" определяют как сходство двух предложений. В оригинальном предложенном алгоритме, сходство предложений вычисляется как нормализованное пересечение этих предложений. Формально:

   $$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \ \& \ w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)}$$

   где S_i - i-ое предложение, состоящее из слов $\{w_k\}$.

2. LexRank [18]. Графовый метод, основанный на TextRank. В отличие от TextRank, этот подход использует иной метод оценки схожести двух предложений для составления весов графа. Все предложения представляются в виде TF-IDF векторов, а сама оценка схожести вычисляется как косинусное сходство, определенное ранее.

3. Latent Semantic Analysis, LSA [19]. Алгебраический метод. Коллекция документов представляется в виде матрицы *количество уникальных терминов в коллекции x количество документов в коллекции.* С помощью сингулярного разложения такой матрицы происходит выбор k предложений.

4. Luhn. Первый известный метод решения задачи суммаризации. Основан на предположении о том, что самые значимые предложения - те, которые содержат наибольшее количество значимых слов(значимые слова - самые часто встречающиеся не стоп-слова)

5. SumBasic [20]. Статистический алгоритм. Основная идея состоит в том, чтобы использовать в документе часто встречающиеся слова, а не менее часто встречающиеся, чтобы получить резюме, которое более вероятно в человеческих рефератах.

6. KL-Sum. В данном алгоритме для выбора краткого содержания S используется следующий критерий:

$$S = min\ KL(P_D||P_S),$$

где $KL(P_D||P_S)$ - расстояние Кульбака - Лейблера между распределениями вероятности появления слов в изначальном документе и в сгенерированном кратком содержании.

Сравнение вышеперечисленных методов проходило на наборе данных CNN, содержащем новостные статьи на английском языке и выдержки (сформированные человеком) из них.

Все методы оценивались по метрике ROUGE [21]. ROUGE - это оценка, которая активно используется для оценивания качества моделей в задачах суммаризации и машинного перевода. Ее идея очень проста: ROUGE-оценка показывает как сильно пересекаются два разных текста. В нашем случае два текста это “правильное” краткое содержание и сгенерированное. В итоговой таблице будут использованы 3 версии ROUGE-оценки: ROUGE-1, ROUGE-2 и ROUGE-L, где число означает количество слов в словосочетании, по которым считается пересечение предложений, а L - означает, что поиск пересечений идет не по фиксированному размеру словосочетаний, а по наибольшему.

Метод\Оценка	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.2343	0.0833	0.1722
LSA	0.2121	0.0705	0.1515
Luhn	0.2397	0.0861	0.1752
LexRank	**0.2531**	**0.0867**	**0.1833**
KL-Sum	0.1936	0.0658	0.1430
SumBasic	0.2463	0.0697	0.1755

Рис. 6: Результаты сравнения методов суммаризации на наборе данных CNN.

На изображении (Рис. 6) отображены результаты сравнения шести алгоритмов на наборе данных CNN. Можно заметить, что лучший результат по ROUGE-оценке показал метод LexRank с оценками ROUGE-1 = 0.2531, ROUGE-2 = 0.0867, ROUGE-L = 0.1833. Это означает, что среди шести данных алгоритмов LexRank будет показывать наилучшие результаты на документах, похожих на используемый для сравнения набор данных CNN.

По результатам сравнения, для составления краткого содержания был выбран алгоритм LexRank.

На изображении (Рис. 7) приведен пример краткого содержания из 10 предложений, полученного из статьи [16]. Исходный текст [16] состоял из 66 предложений.

```
The Only Way to Be Truly Confident in Yourself

Dating or sleeping with more people doesn't necessarily make you feel more confident about how attractive you are.

Moving in with your partner or getting married doesn't necessarily make you feel any more confident in your relatio
nship.

A person confident in their social life will feel as though they lack nothing in their social life.

That you already have, or at least deserve, whatever you feel you would need to make you confident.

After all, at least you're doing something about your lack of confidence.

So here's the real answer: The only way to be truly confident is to simply become comfortable with what you lack.

People who are confident in business are confident because they're comfortable with failure.

People who are confident in their social lives are confident because they're comfortable with rejection.

People who are confident in their relationships are confident because they're comfortable with getting hurt.
```

Рис. 7: Пример работы алгоритма LexRank.

2.2.2 KPMiner

По результатам сравнения (Рис. 2), лучшие результаты показал метод извлечения ключевых фраз KPMiner.

KPMiner (KPM) - статистический алгоритм извлечения ключевых фраз, основанный на мере TF-IDF. Принцип работы KPM совпадает с общим принципом работы методов по извлечению ключевых слов, который был описан в секции Извлечение ключевых слов. Особенностью алгоритма является используемый в нем метод оценки кандидатов. Так, в KPM кандидат с индексом i получает оценку w_i равную:

$$w_i = tf_i * idf_i * B,$$

где B - повышающий фактор, который равен:

$$B = min(\sigma, N_d/(P_d * \alpha)),$$

где σ и α - параметры повышающего фактора (в оригинальной статье: $\sigma = 3$, $\alpha = 2.3$), N_d - общее количество фраз-кандидатов и P_d - общее количество фраз-кандидатов, которые состоят более чем из одного слова.

Таким образом, KPMiner - модификация метода TF-IDF.

Пример работы KMP на статье [16] приведен в Рис. 8. Далее полученные ключевые слова используются для получения набора релевантных изображений.

```
confident
confidence conundrum
loser loop
lack confidence
feel
losers
conundrum
pizza
people
loved
```

Рис. 8: Пример работы алгоритма KPMiner. Представлены 10 ключевых фраз, которые получили наибольшие оценки.

2.2.3 Universal Sentence Encoder

После составления, списки ключевых фраз отправляются в качестве запроса в сервис Pixabay. В ответе приходит набор из изображений (Рис. 9), который удовлетворяет ключевым словам. Далее необходимо для каждого предложения из краткого содержания подобрать подходящее изображение из полученного набора. У каждого из изображений есть текстовое описание. Подбор осуществляется с помощью модели векторизации предложений Universal Sentence Encoder. Для каждого предложения из краткого содержания набор текстовых описаний изображений ранжируется по косинусной схожести их векторных представлений. Таким образом, для текста слайда d выбирается изображение с текстовым описанием a из набора текстовых описаний A такое, что:

$$a = argmax_{a_i \in A}\ CosineSimilarity(USE(d), USE(a_i)),$$

где $USE(x)$ - операция ветктворного представления строки x с помощью Universal Sentence Encoder. Несколько примеров подбора изображений: 11.


```
{'id': 3316342,
 'pageURL': 'https://pixabay.com/illustrations/girl-confident-portrait-cartoon-3316342/',
 'type': 'illustration',
 'tags': 'girl, confident, portrait',
 'previewURL': 'https://cdn.pixabay.com/photo/2018/04/13/11/52/girl-3316342_150.jpg',
 'previewWidth': 150,
 'previewHeight': 150,
 'webformatURL': 'https://pixabay.com/get/g97d4c9704f23ec5d44bc388aed771db364f3bf9f13b170cd2ca810c5816e18b42613959a
7c6722c36acb343b1b89375a88c2c653472ab04b86775aa04afc5789_640.jpg',
 'webformatWidth': 640,
 'webformatHeight': 640,
 'largeImageURL': 'https://pixabay.com/get/g7e68396c0b4881771bf8a3d7b0d05acf4719a34e0efc89f3a86c4cb47680f4c8fde63bc
7ff141f4a2129e5c6cb09440fa91eb33a2ca18138f60c6c5f665aeba7_1280.jpg',
 'imageWidth': 3000,
 'imageHeight': 3000,
 'imageSize': 272269,
 'views': 125985,
 'downloads': 43632,
 'favorites': 587,
 'likes': 608,
 'comments': 92,
 'user_id': 5858294,
 'user': 'lavnatalia',
 'userImageURL': 'https://cdn.pixabay.com/user/2021/01/21/13-20-00-706_250x250.png'}
```


Рис. 9: Пример объекта с изображением. В поле "tags"указано текстовое описание изображения.

2.2.4 XLM-RoBERTa-xnli

Последний этап - определение тематики музыкального сопровождения (Классификация без подготовки). Для задачи zero-shot классификации была выбрана предобученная модель XLM-RoBERTa-xnli. XLM-RoBERTa [22] - мультиязыковая модель, обученная, в общей сложности, на 2,5 терабайта данных на 100 различных языках. Эта модель было предложена в [22] и использует ту же архитектуру, что и RoBERTa [23]. XLM-RoBERTa-xnli - это XLM-RoBERTa дообученная на наборе данных xnli (мультиязыковой набор данных для задачи NLI, описанной в Классификация без подготовки).

По умолчанию, в разработанном сервисе используются следующие возможные тематики музыкального сопровождения: sad (грустная), happy (веселая), inspirational (воодушевляющая), motivational (мотивирующая), relaxing (успокаивающая), news (новости), business (бизнес), kids (дети), sport (спорт). Однако, как было сказано ранее, модель не привязана к определенному набору классов, поэтому пользователь может свободно корректировать набор классов.

Несколько примеров приведены в Рис 10. На вход функции подается заголовок статьи или любой другой отрывок, а на выходе - заданные классы, вместе с оценками $P(h|p)$ (была определена в Классификация без подготовки), отсортированные по убыванию. Таким образом, первый класс можно использовать как тематику музыкального сопровождения для видеопрезентации.

```
Ввод [36]: get_audio_theme('The Only Way to Be Truly Confident in Yourself')

Out[36]: [('motivational', 0.47767549753189087),
          ('inspirational', 0.17200331389904022),
          ('relaxing', 0.09099747985601425),
          ('happy', 0.0710340216755867),
          ('news', 0.07007536292076111),
          ('business', 0.05248863995075226),
          ('sport', 0.023927368223667145),
          ('sad', 0.022060681134462357),
          ('kids', 0.019737739115953445)]

Ввод [33]: get_audio_theme("История одного успешного футболиста")

Out[33]: [('sport', 0.8487090468406677),
          ('inspirational', 0.10309497267007828),
          ('motivational', 0.018715327605605125),
          ('happy', 0.007924054749310017),
          ('news', 0.006504741497337818),
          ('business', 0.0049812220968306065),
          ('relaxing', 0.0048119486309587955),
          ('kids', 0.004176552407443523),
          ('sad', 0.0010822577169165015)]
```

Рис. 10: Примеры работы модели XLM-RoBERTa-xnli на русском и английском языках.

Заключение

2.3 Результаты

В данной работе была решена задача разработки сервиса по автоматическому созданию видеопрезентаций из текста. В ходе работы был проведен обзор предметной области. Были рассмотрены такие задачи обработки естественного языка, как:

- Text Summarization
- Keywords Extraction
- Text Similarity
- Zero-shot Text Classification

Для выполнения каждой из подзадач сервиса были выбраны модели, которые имеют наилучшее качество, устойчивость к смене домена и устойчивость к смене языка исходного текста. В результате, сервис, сохраняя высокое качество, способен поддерживать 16 языков. Сервис может поддерживать более 100 языков при условии замены Universal Sentence Encoder компоненты на TF-IDF.

В результате объединения всех компонент, на фреймворке Flask был разработан API-сервис. С его помощью можно получить все необходимое содержимое для составления видеопрезентации. API-сервис можно интегрировать в уже существующие редакторы видео, чтобы получить готовый видеоформат.

2.4 Перспективы развития

Перспективы развития проекта следующие:

- Интегрирование сервис в существующие инструменты по редактированию видео.

- Использование генеративных моделей, например, для генерации фона слайда по текстовому содержанию слайда.
- Расширение функционала сервиса.

Список литературы

[1] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[4] Eirini Papagiannopoulou and Grigorios Tsoumakas. A review of keyphrase extraction. *CoRR*, abs/1905.05044, 2019.

[5] Mikalai Krapivin, Aliaksandr Autayeu, and Maurizio Marchese. Large dataset for keyphrases extraction. Technical Report DISI-09-055, DISI, Trento, Italy, May 2008. http://eprints.biblio.unitn.it/archive/00001671/01/disi09055-krapivin-autayeu-marchese.pdf.

[6] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[7] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Torvik Sølvberg, and Edie Rasmussen, editors, *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[8] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical*

Methods in Natural Language Processing, EMNLP '03, page 216–223, USA, 2003. Association for Computational Linguistics.

[9] Luís Marujo, Márcio Viveiros, and João Paulo da Silva Neto. Keyphrase cloud generation of broadcast news. *CoRR*, abs/1306.4606, 2013.

[10] Samhaa R. El-beltagy. Kp-miner: A simple system for effective keyphrase extraction. In *2006 Innovations in Information Technology*, pages 1–5, 2006.

[11] Quoc V. Le and Tomás Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[14] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[15] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, 2019.

[16] Mark Manson. The only way to be truly confident in yourself, 2021.

[17] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[18] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128, 2011.

[19] Makbule Ozsoy, Ferda Alpaslan, and Ilyas Cicekli. Text summarization using latent semantic analysis. *J. Information Science*, 37:405–417, 08 2011.

[20] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. Technical report, Microsoft Research, 2005.

[21] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[22] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Приложения

'The Only Way to Be Truly Confident in Yourself'

'After all, at least you're doing something about your lack of confidence.'

'Телефонный мошенник на меня обиделся за наглый, грубый смех'

'Богатые экономят на том, на что бедные постоянно тратят свои деньги'

'Богатый человек знает, что чем больше денег он вкладывает в себя, тем больше он сможет заработать.'

Рис. 11: Примеры изображений подобранных к текстовому описанию слайда.