MISIS AI LAB

ALPHA

Автоматизация работы с документами посредством введения нейронных сетей

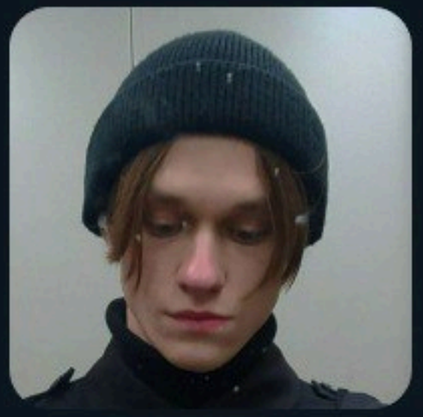
Наша команда

Савельев Ярослав
Robotics engineer -
Sber Robotics Lab

Комлев Данила
Frontend/Backend - INVITRO

Иванов Арсений
Исследователь данных -
Sber AmazMe

Волков Даниил
MLOps engineer -
ГосУслуги



Максим
ML engineer -
Точка Банк

ПОСТРОЕНИЕ ИНТРЕПРЕТИРУЕМОСТИ

BOW/TF-IDF НА Н-ГРАММАХ ПО СЛОВАМ + ДЕРЕВЬЯ/ЛОГРЕГ ->

ПЛЮСЫ: ЛЕГКО ИНТЕРПРЕТИРОВАТЬ И ЗАХВАТЫВАТЬ ВЕСЬ ДОКУМЕНТ
МИНУСЫ: КЛЮЧЕВЫЕ СЛОВА НЕПОНЯТНЫ В КОНТЕКСТЕ

BERT ПО ПЕРВЫМ 512 ТОКЕНАМ ДОКУМЕНТА + ГРАДИЕНТНЫЕ/ ATTENTION МЕТОДЫ ИНТЕРПРЕТАЦИИ ->

ПЛЮСЫ: БЫСТРО РЕАЛИЗОВАТЬ
МИНУСЫ: НЕУДОБНО ЗАНОВО ПРОСМАТРИВАТЬ ДОКУМЕНТ, УЧИТЫВАЕМ НЕ ВЕСЬ ТЕКСТ ДОКУМЕНТА, ТЯЖЕЛЫЙ ИНФЕРЕНС

BERT ПО ФРАГМЕНТАМ ДОКУМЕНТА + УСРЕДНЕНИЕ СКОРА ФРАГМЕНТОВ ->

ПЛЮСЫ: УДОБНО ИНТЕРПРЕТИРОВАТЬ, УЧИТЫВАЕМ ВЕСЬ ДОКУМЕНТ
МИНУСЫ: ТРУДНО РЕАЛИЗОВАТЬ, ТЯЖЁЛЫЙ ИНФЕРЕНС

МЕТРИКИ

		1 class	2 class	3 class	4 class	5 class	macro
	recall						0.97
BERT	f1-score						0.98
	precision						0.98
	recall						0.96
TFiDF + CatBoost	f1-score						0.97
	precision						0.97

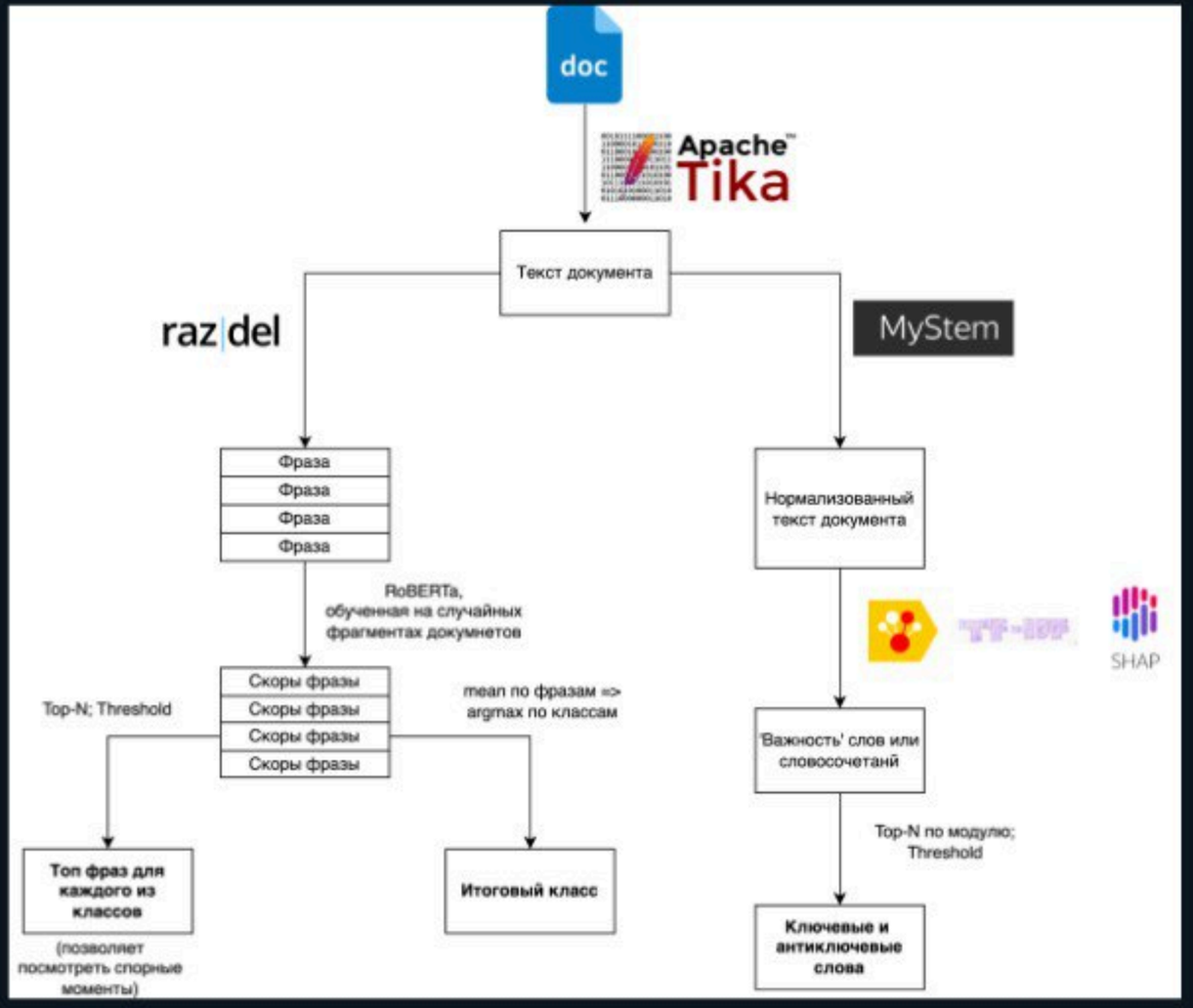
	accuracy		
BERT	0.98		
TFiDF + CatBoost	0.97		

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Precision = \frac{\sum TruePositive}{\sum TruePositive + FalsePositive}$$

$$Recall = \frac{\sum TruePositive}{\sum TruePositive + FalseNegative}$$

ИНФЕРЕНС


doc
Apache Tika
Текст документа
raz|del
MyStem
Фраза
Фраза
Фраза
Фраза
RoBERTa, обученная на случайных фрагментах документов
Нормализованный текст документа
SHAP
Скоры фразы
Скоры фразы
Скоры фразы
Скоры фразы
Top-N; Threshold
mean по фразам => argmax по классам
'Важность' слов или словосочетаний
Top-N по модулю; Threshold
Топ фраз для каждого из классов
(позволяет посмотреть спорные моменты)
Итоговый класс
Ключевые и антиключевые слова

МАСШТАБИРОВАНИЕ МОДЕЛИ

Published as a conference paper at ICLR 2020

DIVIDEMIX: LEARNING WITH NOISY LABELS AS SEMI-SUPERVISED LEARNING

Junnan Li, Richard Socher, Steven C.H. Hoi
Salesforce Research
{junnan.li,rsocher,shoi}@salesforce.com

ABSTRACT

Deep neural networks are known to be annotation-hungry. Numerous efforts have been devoted to reducing the annotation cost when learning with deep networks. Two prominent directions include learning with noisy labels and semi-supervised learning by exploiting unlabeled data. In this work, we propose DivideMix, a novel framework for learning with noisy labels by leveraging semi-supervised learning techniques. In particular, DivideMix models the per-sample loss distribution with a mixture model to dynamically divide the training data into a labeled set with clean samples and an unlabeled set with noisy samples, and trains the model on both the labeled and unlabeled data in a semi-supervised manner. To avoid confirmation bias, we simultaneously train two diverged networks where each network uses the dataset division from the other network. During the semi-supervised training phase, we improve the MixMatch strategy by performing label co-refinement and label co-guessing on labeled and unlabeled samples, respectively. Experiments on multiple benchmark datasets demonstrate substantial improvements over state-of-the-art methods. Code is available at https://github.com/LiJunnan1992/DivideMix.

1 INTRODUCTION

The remarkable success in training deep neural networks (DNNs) is largely attributed to the collection of large datasets with human annotated labels. However, it is extremely expensive and time-consuming to label extensive data with high-quality annotations. On the other hand, there exist alternative and

Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching

Liu Yang Mingyang Zhang Cheng Li Michael Bendersky Marc Najork
Google Research, Mountain View, CA, USA
{yangliuy,mingyang,chgli,bemike,najork}@google.com

CT

al language processing and information retrieval prob-formalized as the task of semantic matching. Existing area has been largely focused on matching between e.g., question answering), or between a short and a ., ad-hoc retrieval). Semantic matching between long-nts, which has many important applications like news ation, related article recommendation and document relatively less explored and needs more research effort. rs, self-attention based models like Transformers [30] 5] have achieved state-of-the-art performance in the matching. These models, however, are still limited to ke a few sentences or one paragraph due to the qua-utational complexity of self-attention with respect to ngth. In this paper, we address the issue by proposing Multi-depth Transformer-based Hierarchical (SMITH) long-form document matching. Our model contains vations to adapt self-attention models for longer text opose a transformer based hierarchical encoder to cap-ment structure information. In order to better capture el semantic relations within a document, we pre-train ith a novel masked sentence block language modeling ion to the masked word language modeling task used r experimental results on several benchmark datasets n document matching show that our proposed SMITH rforms the previous state-of-the-art models including attention [34], multi-depth attention-based hierarchi-t neural network [14], and BERT. Comparing to BERT es, our model is able to increase maximum input text 512 to 2048. We will open source a Wikipedia based lataset, code and a pre-trained checkpoint to accelerate rch on long-form document matching.[1]

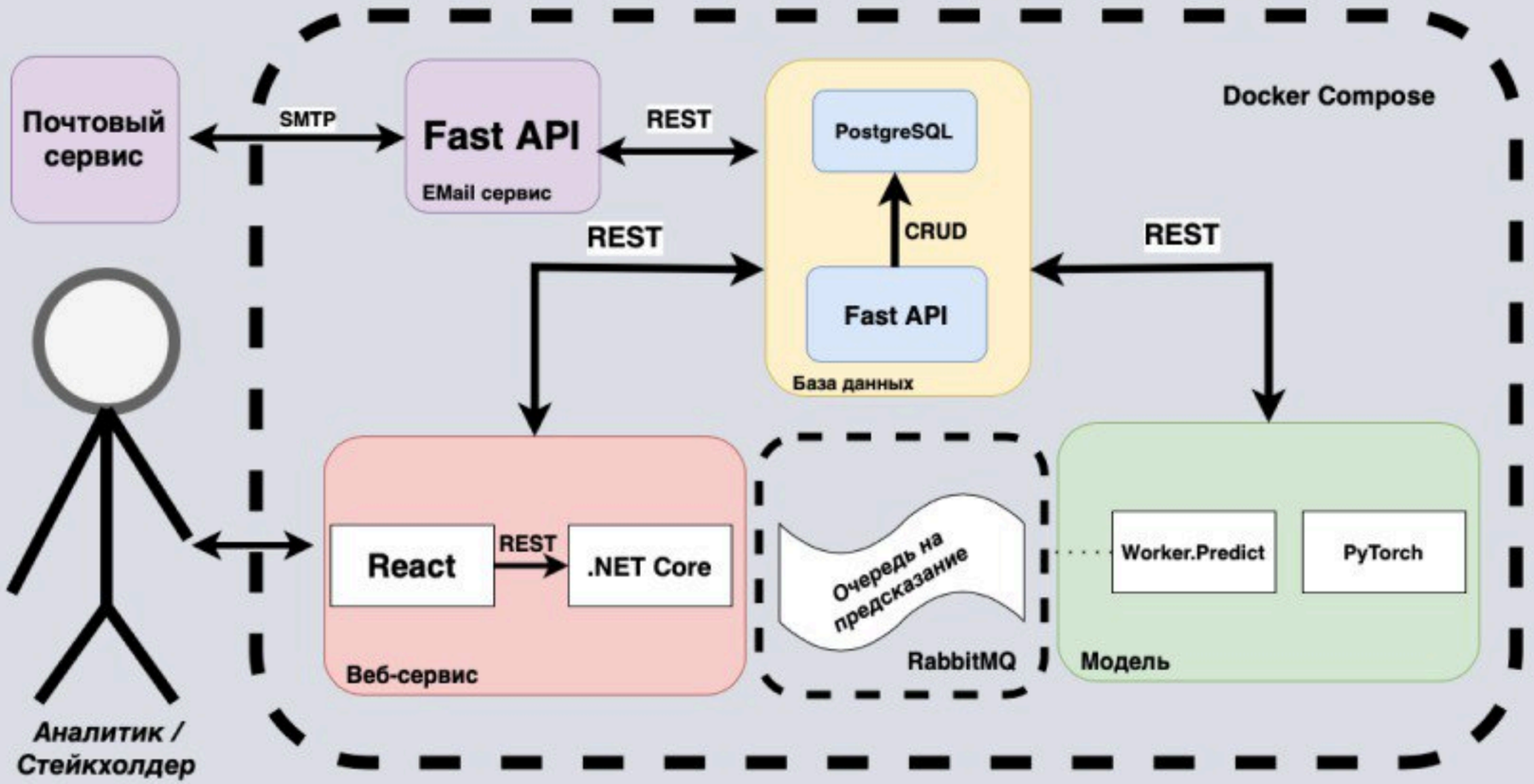
nce Format:
lingyang Zhang Cheng Li Michael Bendersky Marc
Hierarchical Encoder for Long-Form Document Matching. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411908

1 INTRODUCTION

Semantic matching is an essential task for many natural language processing (NLP) and information retrieval (IR) problems. Research on semantic matching can potentially benefit a large family of applications including ad-hoc retrieval, question answering and recommender systems [17]. Semantic matching problems can be classified into four different categories according to text length, including short-to-short matching, short-to-long matching, long-to-short matching and long-to-long matching. Table 1 shows a classification of different semantic matching tasks with example datasets. Semantic matching between short text pairs is relatively well studied in previous research on paraphrase identification [38], natural language inference [1], answer sentence selection [35], etc. Short-to-long semantic matching like relevance modeling between query/ document pairs has also been a popular research topic in IR and NLP communities [3]. For long-to-short semantic matching, there are also a variety of research on tasks like conversation response ranking, which is to match a conversation context with response candidates [18]. To the best of our knowledge, semantic matching between long document pairs, which has many important applications like news recommendation, related article recommendation and document clustering, is less explored and needs more research effort. Table 2 shows an example of semantic matching between document pairs from Wikipedia. These documents have thousands of words organized in sections, passages and sentences.

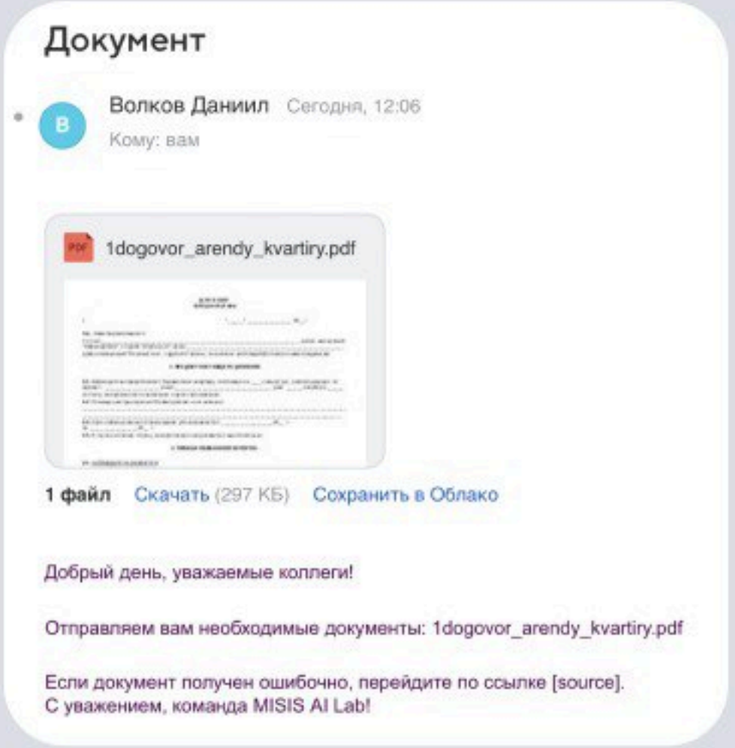
Compared to semantic matching between short texts, or between short and long texts, semantic matching between long texts is a more challenging task due to a few reasons: 1) When both texts are long, matching them requires a more thorough understand-ing of semantic relations including matching pattern between text

BACKEND

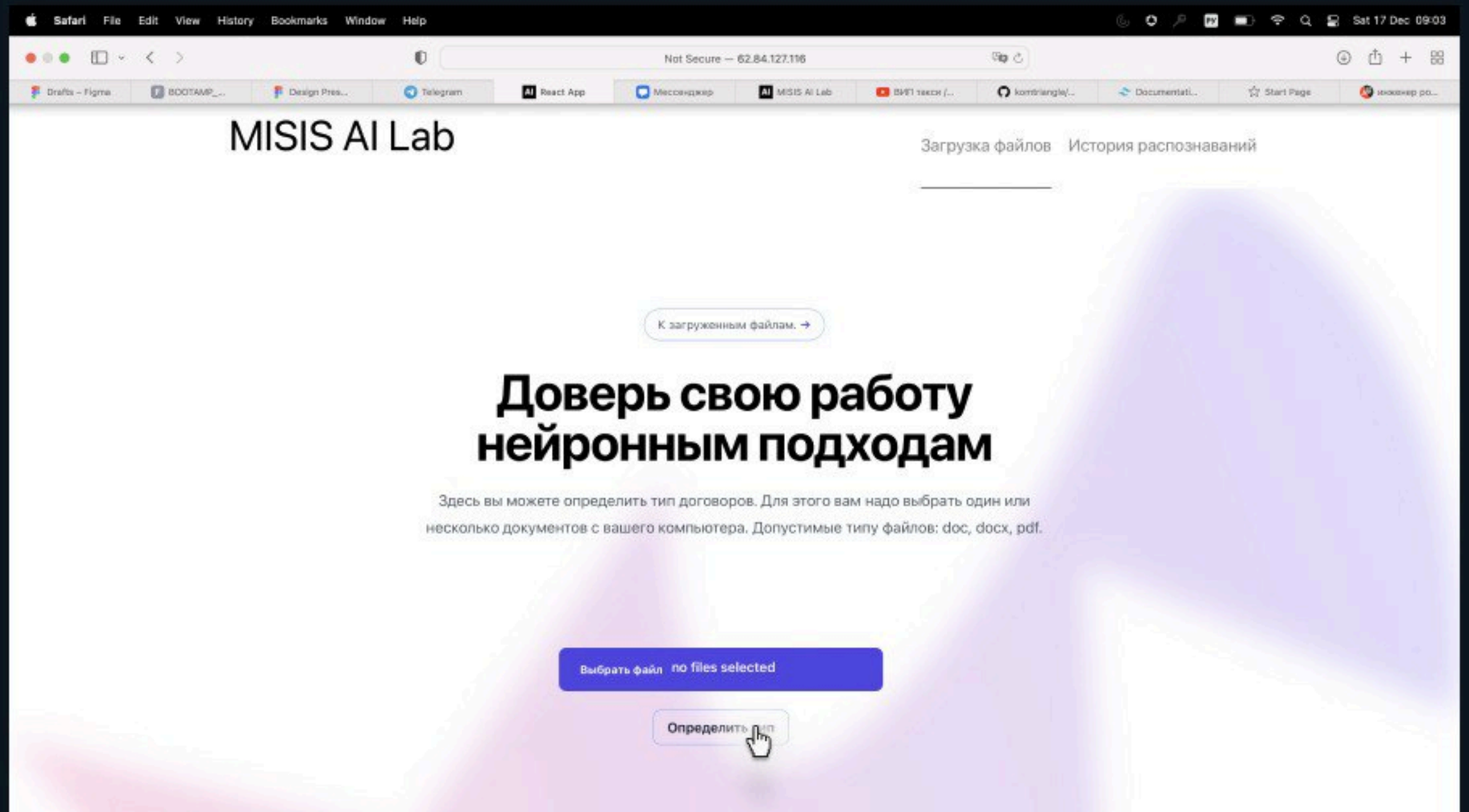

Почтовый сервис
SMTP
Fast API
EMail сервис
REST
PostgreSQL
CRUD
Fast API
База данных
Docker Compose
REST
REST
React
REST
.NET Core
Веб-сервис
Очередь на предсказание
RabbitMQ
Worker.Predict
PyTorch
Модель
Аналитик / Стейкхолдер

FEATURES

В базе данных хранится сопоставление типов документов и емэйлов. После определения типа документа, данный файл будет отправлен на все емейлы, которые привязаны к этому типу документа.



React JS Redux Sass



THE END