**Generating Summary for Terms and Conditions Document Based on the Query provided.**

**Abstract**

We introduce a system for summary generation based on the terms of service documents we usually agree to by reading the entire or useful content. Generally, we skip reading large documents and end up agreeing to some things that we don't like to do. What if we have a system that can summarize the piece of information that you are worried about in a large document and get the important sentences? Our system takes the URL of these terms and conditions documents and the query we are concerned about as the input and produces the summarized sentences.

**Introduction**

Usually when we come across the terms and conditions documents most of the users just simply click on agree button and move on rather than reading the entire document. And some people might search for some points like is their data is confidential or not and to do so they have to search the entire document and finally they may find the information that is relevant but they have to read the information and again fetch the important points. Despite of doing all these tasks there is a chance of missing the overall objective of the information that we see.

What if we have a system which can take the url of the online document that we are referring too and the focused area as input and give us the sentences which covers the area that we need to focus on. If we do so we dont need to worry about reading the entire document or missing any important points.

**Problem**

In our system we have two inputs one is the document that we are currently dealing with In our project we are summarizing the terms and conditions documents which mainly talks about the privacy of the data. And we are experimenting on mailny three documents those are mentioned below.

**Input:**

* [**https://terms.account.samsung.com/contents/legal/usa/eng/general.html**](https://terms.account.samsung.com/contents/legal/usa/eng/general.html)
* [**https://www.apple.com/legal/privacy/data/en/apple-id/**](https://www.apple.com/legal/privacy/data/en/apple-id/)
* [**https://s3.amazonaws.com/oneplussupport/Servify/Terms+&+Conditions+-+OnePlus+Secure+by+Servify+update.pdf**](https://s3.amazonaws.com/oneplussupport/Servify/Terms+&+Conditions+-+OnePlus+Secure+by+Servify+update.pdf)

**Area of focus:**

* Data, user data, confidentiality, private information

And based on the provided input document and the area of interest we will be fetching out the sentences that provide the useful information.

**Output:**

Output is the summarized text based on the document and the initial query provided by the user.

**Sample output** :- When you upload, transmit, create, post, display, or otherwise provide any information, materials, documents, media files, or other content on or through our Services, you grant us an irrevocable, unlimited, worldwide, royalty-free, and non-exclusive license to copy, reproduce, adapt, modify, edit, distribute, translate, publish, publicly perform, and publicly display such User Content to the full extent allowed by Applicable Law.

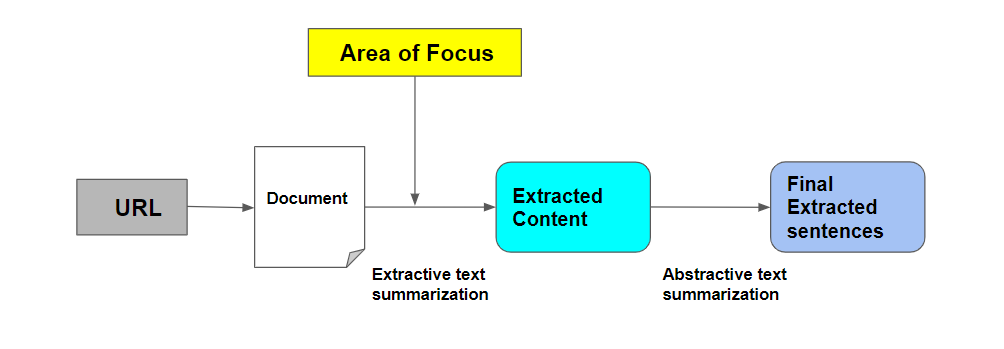
**Related Work**

In the text summarization there are two types of summarizations. They are extractive and abstractive text summarizations. These is lot of work has focused on Extractive summarization. Extractive summarization takes the original text and extracts information that is identical to it. In other words, rather than providing a unique summary based on the full content ((Neto et al., 2002), (Erkan and Radev, 2004), (Filippova and Altun, 2013), (Colmenares et al., 2015), (Riedhammer et al., 2010), (Ribeiro et al., 2013)). There has been some work on abstractive summarization in the context of DUC-2003 and DUC-2004 contests (Zajic et al.). We refer the reader to (Das and Martins, 2007) and (Nenkova and McKeown, 2012) for an excellent survey of the field.

**Approach**

In our problem we cannot directly perform the text summarization to get the sentences based on the area of focus we have to first get the relevant data or passages from the long documents that we use as an input based on the area of focus that we provide and ask for.

After getting the content from the extractive text summarization we provide this content as an input and perform abstractive text summarization on this input and finally we get our desired sentences.

****

**Readibility metrics**

Before getting into the actual data pre processing and summarization. We are using some readability metrics to the raw data in the document. These readability metrics helps us to understand the ease of how a user can read the actual document. It depends on the content in the document and the size of the document.

Here are some results of four types of readability metrics that we used to evaluate the document.

| flesch\_kincaid | gunning\_fog | dale\_chall | flesch |
| --- | --- | --- | --- |
| grade\_level: '14' | grade\_level: 'college' | grade\_levels: ['college\_graduate'] | ease: 'difficult', grade\_levels: ['college'] |

We can clearly observe that for each readability metric here the level of difficulty is above the grad level which is difficult to read.

**Data Pre Processing**

**Our data pre processing consists of the following steps**

* Input data
* Importing libraries
* Stopword removal
* Word Tokenization

In our pre processing stage we import the important libraries. We import spacy which is a free open source library for advanced Natural Language Processing. From Spacy we are importing STOP\_WORDS and from string class we are importing class punctuation. To load the models and data for the English Language, we are using spacy.load(‘en\_core\_web\_sm’) which is a english language model which is loaded via spacy. We are providing the content to the NLP object which is used as a pre-processing pipeline. All the tokenizing, parsing and pre processing activities will be taken care by using this object. THe NLP object returns an other object that holds the information of the tokens and their relations.

After which we will perform the extractive text summarization to get the extracted content based on the query that we provide and also perform abstractive summarization. To get

**Evaluation**

**Discussion**

**Conclusion**

**References**

* [**https://www.sciencedirect.com/science/article/abs/pii/S0885230820300991**](https://www.sciencedirect.com/science/article/abs/pii/S0885230820300991)
* [**https://blog.agolo.com/query-based-summarization-in-action-ea729df3109c**](https://blog.agolo.com/query-based-summarization-in-action-ea729df3109c)
* [**https://ieeexplore.ieee.org/document/7557434**](https://ieeexplore.ieee.org/document/7557434)