2019 EE614

Service Oriented Computing

Term Project

Web service deploying deep learning model

2019 Vahn

20194334 Soyeon Kim

1. Introduction
   1. Motivation

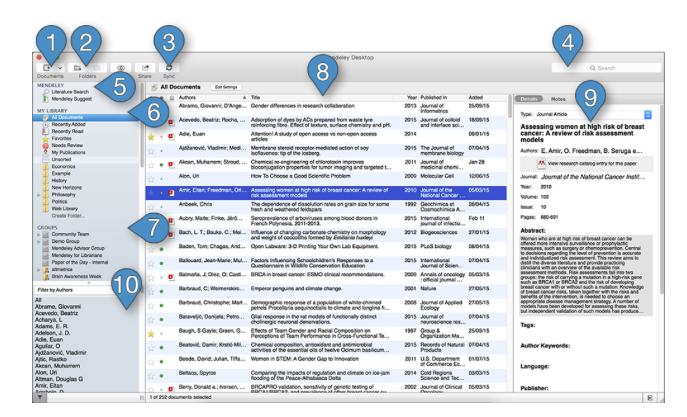
We are exposed to many occasions to show our ideas, share what we’ve done or write and submit the project what we have done. Under whatsoever situations, we put many pictures in our documents to make our idea seem more precise, clear to the audience or readers. Usually under the pictures in documents, proper descriptions are required such as the main idea of picture or source where we got these resources from. The point is that it is so tedious to put all the descriptions on our own.

Thus, we propose a web service system which can generate appropriate captions for pictures in users’ documents. We think this service can have more impact when integrated to current programs such as MS office, google docs and so on.

* 1. Existing Service

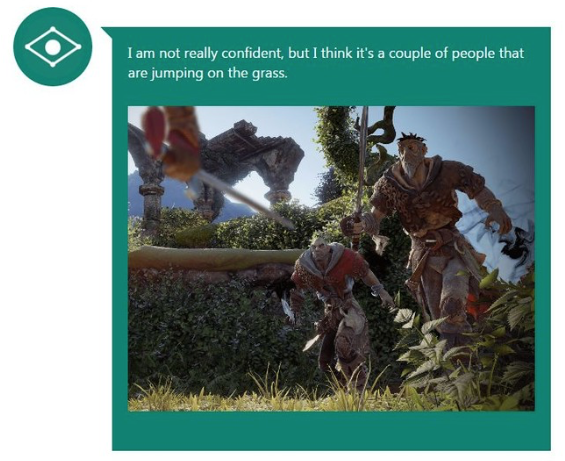
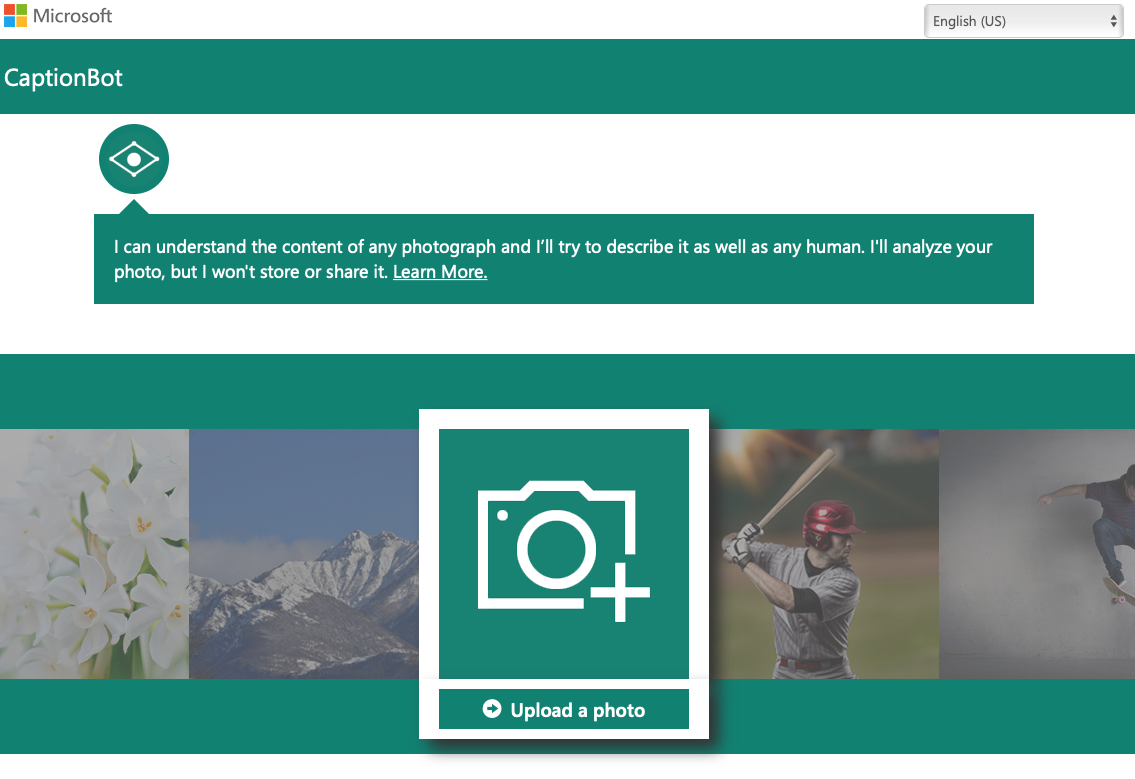
1. Mendeley

Mendeley is a company based in London, UK, which provides products and services for academic researchers. It is most known for its [reference manager](https://en.wikipedia.org/wiki/Reference_management) which is used to manage and share research papersand generate [bibliographies](https://en.wikipedia.org/wiki/Bibliography) for scholarly articles. Even this service is not directly related to the images in the documents, we can infer that managing reference which is also a sort of tedious works in academic area is a huge fascination to the people.



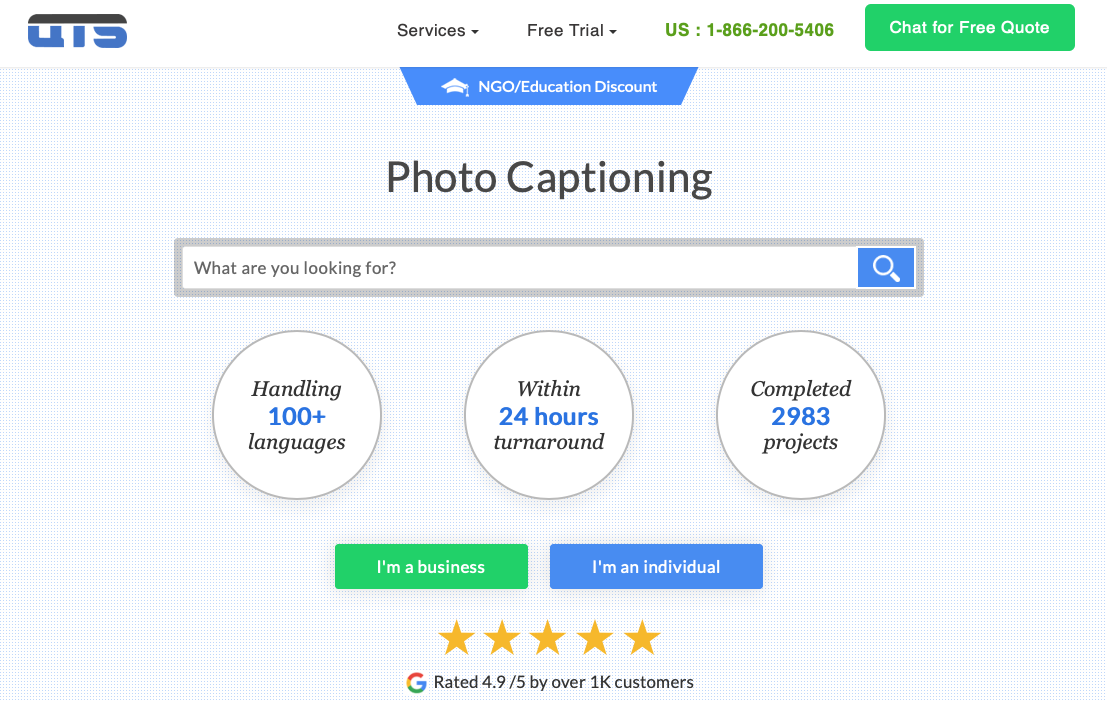
1. Caption bot

CaptionBot.ai is powered by machine learning technology that identifies and captions your photos. This comes from Microsoft Cognitive Services, the company's division that created the APIs that are used for the previously launched [HowOld.net](https://www.windowscentral.com/microsoft-trying-guess-how-old-you-look-new-website-and-face-detection-api) or [TwinorNot.com](http://TwinorNot.com) websites. In this case, the Captionbot site uses Microsoft's Computer Vision, Emotion and Bing Image APIs to come up with the captions for the uploaded photos. With this service, we can find image translating service is near our hands.



1. Quick transcription service

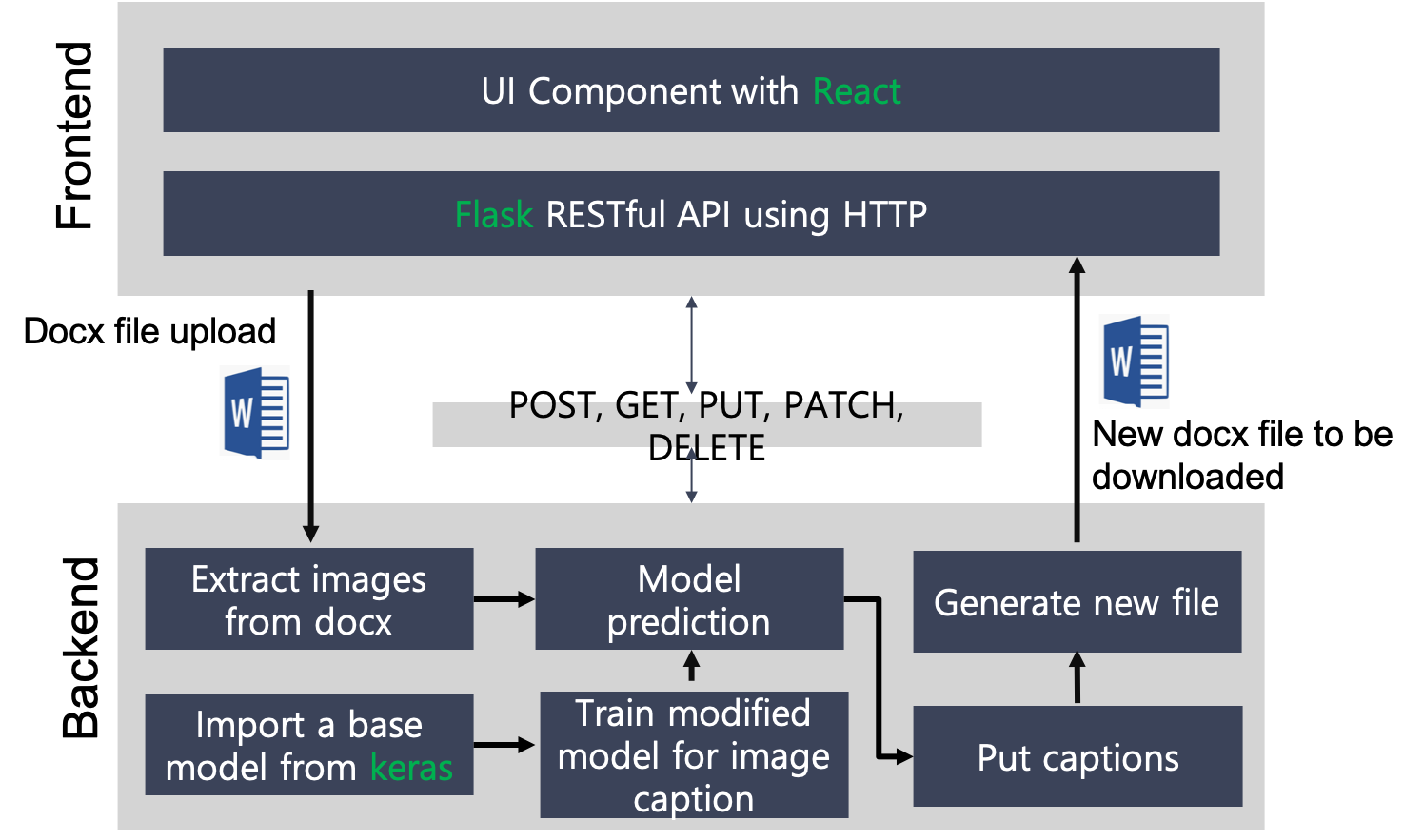
Photo Captioning is a process of creating captions for images. When customers search for images through search engines it is those captions that are mapped to the keyword that are being searched and pulls up your image. Quick Transcription Service helps in all possible ways to get users images sold. Once the customer can bring it up through search engines then the quality of the photo sells for itself. This service shows that translating image can be served as a service.



1. Proposed solution

We decide to make this service as a web application first using our knowledge based on the lecture provided in this course. The overall service architecture is below. The service is mainly separate in two parts:

1. Web framework: front-end is where user actually interact with the service. React.js provides simple and user-friendly web design interface. Flask is a flexible micro web framework written in Python and controls the back-end. More technical explanation will be followed in section 4.
2. Deep learning model: when the user sends the file, the back-end with pre-trained model needs to predict and produce the result. Because our goal of this project is to make a caption of the image, we use an image captioning algorithm. More details are described in section3.



Besides above parts, we use python library(doc2text) to extract the image that needs to be analyzed from the documents and to put captions(docxtpl).

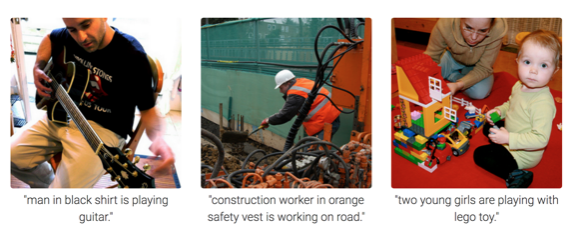
1. Deep learning model

3-1. Image captioning

Every day, we encounter a large number of images from various sources such as the internet, news articles, document diagrams and advertisements. These sources contain images that viewers would have to interpret themselves. Most images do not have a description, but the human can largely understand them without their detailed captions. However, machine needs to interpret some form of image captions if humans need automatic image captions from it.

Image captioning is important for many reasons. For example, they can be used for automatic image indexing. Image indexing is important for Content-Based Image Retrieval (CBIR) and therefore, it can be applied to many areas, including biomedicine, commerce, the military, education, digital libraries, and web searching. Social media platforms such as Facebook and Twitter can directly generate descriptions from images. The descriptions can include where we are (e.g., beach, cafe), what we wear and importantly what we are doing there.

Image captioning is a popular research area of Artificial Intelligence (AI) that deals with image understanding and a language description for that image. Image understanding needs to detect and recognize objects. It also needs to understand scene type or location, object properties and their interactions. Generating well-formed sentences requires both syntactic and semantic understanding of the language. Understanding an image largely depends on obtaining image features. The techniques used for this purpose can be broadly divided into two categories: (1) Traditional machine learning based techniques and (2) Deep machine learning based techniques.



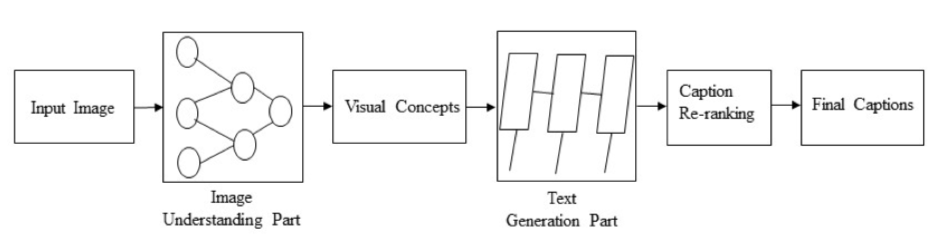
In traditional machine learning, hand crafted features such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), the Histogram of Oriented Gradients (HOG) , and a combination of such features are widely used. In these techniques, features are extracted from input data. They are then passed to a classifier such as Support Vector Machines (SVM) in order to classify an object. Since hand crafted features are task specific, extracting features from a large and diverse set of data is not feasible. Moreover, real world data such as images and video are complex and have different semantic interpretations.

On the other hand, in deep machine learning based techniques, features are learned automatically from training data and they can handle a large and diverse set of images and videos. For example, Convolutional Neural Networks (CNN) are widely used for feature learning, and a classifier such as Softmax is used for classification. CNN is generally followed by Recurrent Neural Networks (RNN) in order to generate captions.

3-2. Deep learning based image captioning

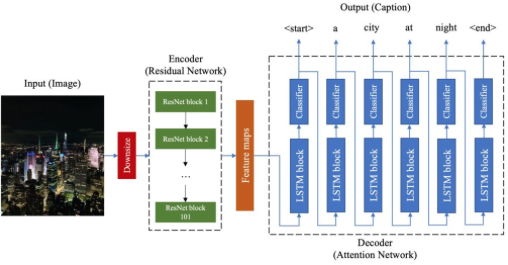
Captions can be retrieved from visual space and multimodal space. In retrieval-based approaches, captions are retrieved from a set of existing captions. Retrieval based methods first find the visually similar images with their captions from the training data set. These captions are called candidate captions. The captions for the query image are selected from these captions pool. These methods produce general and syntactically correct captions. However, they cannot generate image specific and semantically correct captions.

Novel captions can be generated from both visual space and multimodal space. A general approach of this category is to analyze the visual content of the image first and then generate image captions from the visual content using a language model. These methods can generate new captions for each image that are semantically more accurate than previous approaches. Most novel caption generation methods use deep machine learning based techniques. Novel caption generation-based image caption methods mostly use visual space and deep machine learning based techniques. Captions can also be generated from multimodal space. Deep learning-based image captioning methods can also be categorized on learning techniques: Supervised learning, Reinforcement learning, and Unsupervised learning. Usually captions are generated for a whole scene in the image. However, captions can also be generated for different regions of an image(Dense captioning). Image captioning methods can use either simple Encoder-Decoder architecture or Compositional architecture. There are methods that use attention mechanism, semantic concept, and different styles in image descriptions. Some methods can also generate description for unseen objects. Most of the image captioning methods use LSTM as language model. However, there are a number methods that use other language models such as CNN and RNN.



3-2-1. Visual Space vs. Multimodal Space

Deep learning-based image captioning methods can generate captions from both visual space and multimodal space. Understandably image captioning datasets have the corresponding captions as text. In the visual space-based methods, the image features and the corresponding captions are independently passed to the language decoder. In contrast, in a multimodal space case, a shared multimodal space is learned from the images and the corresponding caption-text. This multimodal representation is then passed to the language decoder. We adopt visual space- based models rather than multimodal space.



3-3. Encoder-Decoder architecture based model description

Most of the models rely on the widespread encoder–decoder framework, which is flexible and effective. Sometimes it is defined as a structure of CNN + RNN. Usually a convolutional neural network (CNN) represents the encoder, and a recurrent neural network (RNN) the decoder. The encoder is the one which “reads” an image—given an input image, it extracts a high-level feature representation. The decoder is the one which generates words—given the image representation from the encoder (encoded image), it generates words to represent the image with a full grammatically and stylistically correct sentence.

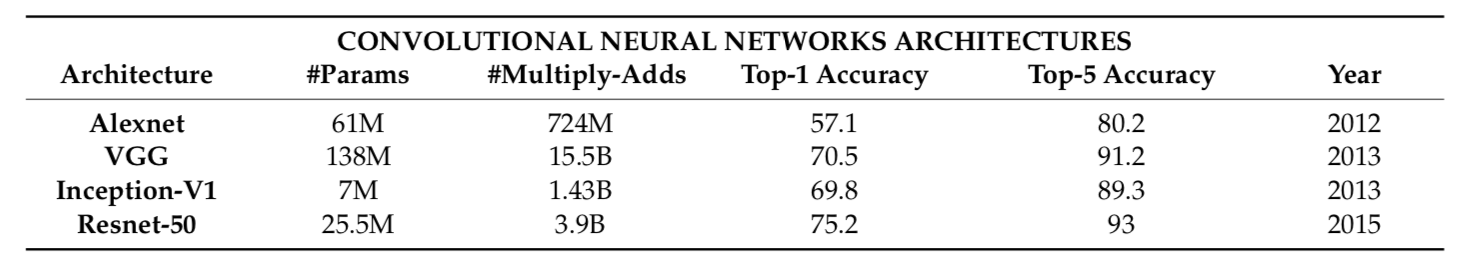
These methods are very similar to the encoder-decoder framework-based neural machine translation. In this network, global image features are extracted from the hidden activations of CNN and then fed them into an LSTM to generate a sequence of words. A typical method of this category has the following general steps: (1) A vanilla CNN is used to obtain the scene type, to detect the objects and their relationships. (2) The output of Step is used by a language model to convert them into words, combined phrases that produce an image captions. CNN for image representations and an LSTM for generating image captions. To prove performance, a trial of use batch normalization in CNN and the output of the last hidden layer of CNN is used as an input to the LSTM decoder. This LSTM is capable of keeping track of the objects that already have been described using text. NIC is trained based on maximum likelihood estimation.

In generating image captions, image information is included to the initial state of an LSTM. The next words are generated based on the current time step and the previous hidden state. This process continues until it gets the end token of the sentence. Since image information is fed only at the beginning of the process, it may face vanishing gradient problems. The role of the words generated at the beginning is also becoming weaker and weaker. Therefore, LSTM is still facing challenges in generating long length sentences fundamentally without scaling its capacity with more hidden layers.

An improved model is proposed an extension of LSTM called guided LSTM (gLSTM). This gLSTM can generate long sentences. In this architecture, it adds global semantic information to each gate and cell state of LSTM. It also considers different length normalization strategies to control the length of captions. Semantic information is extracted in different ways. First, it uses a cross-modal retrieval task for retrieving image captions and then semantic information is extracted from these captions. The semantic based information can also be extracted using a multimodal embedding space.

3-3-1. Encoder – CNN

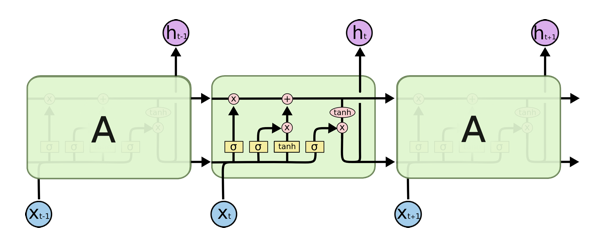
As there is usually only one encoder in the model, the performance is highly reliant on the CNN deployed. Even though we identified five convolutional networks in our research, there are two which stand out and were used the most. The first most popular choice for the feature extractor from images is VGGNet, preferred for the simplicity of the model and for its power. Even the performance of VGGNet is powerful, however, the same number of articles which used ResNet as an encoder was also found. ResNet wins for being computationally the most efficient compared to all other convolutional networks. In following table, a clear comparison of four networks—AlexNet, VGGNet, ResNet, and GoogleNet (also called Inception-X Net) is provided



3-3-2. Decoder—LSTM

LSTM (long-short-term memory) was developed from RNN, with the intention to work with sequential data. It is now considered as the most popular method for image captioning due to its effectiveness in memorizing long term dependencies through a memory cell. Undoubtedly this requires a lot of storage and is complex to build and maintain. LSTM works by generating a caption by making one word at every time step conditioned on a context vector, together with the previous hidden state and the earlier generated words.

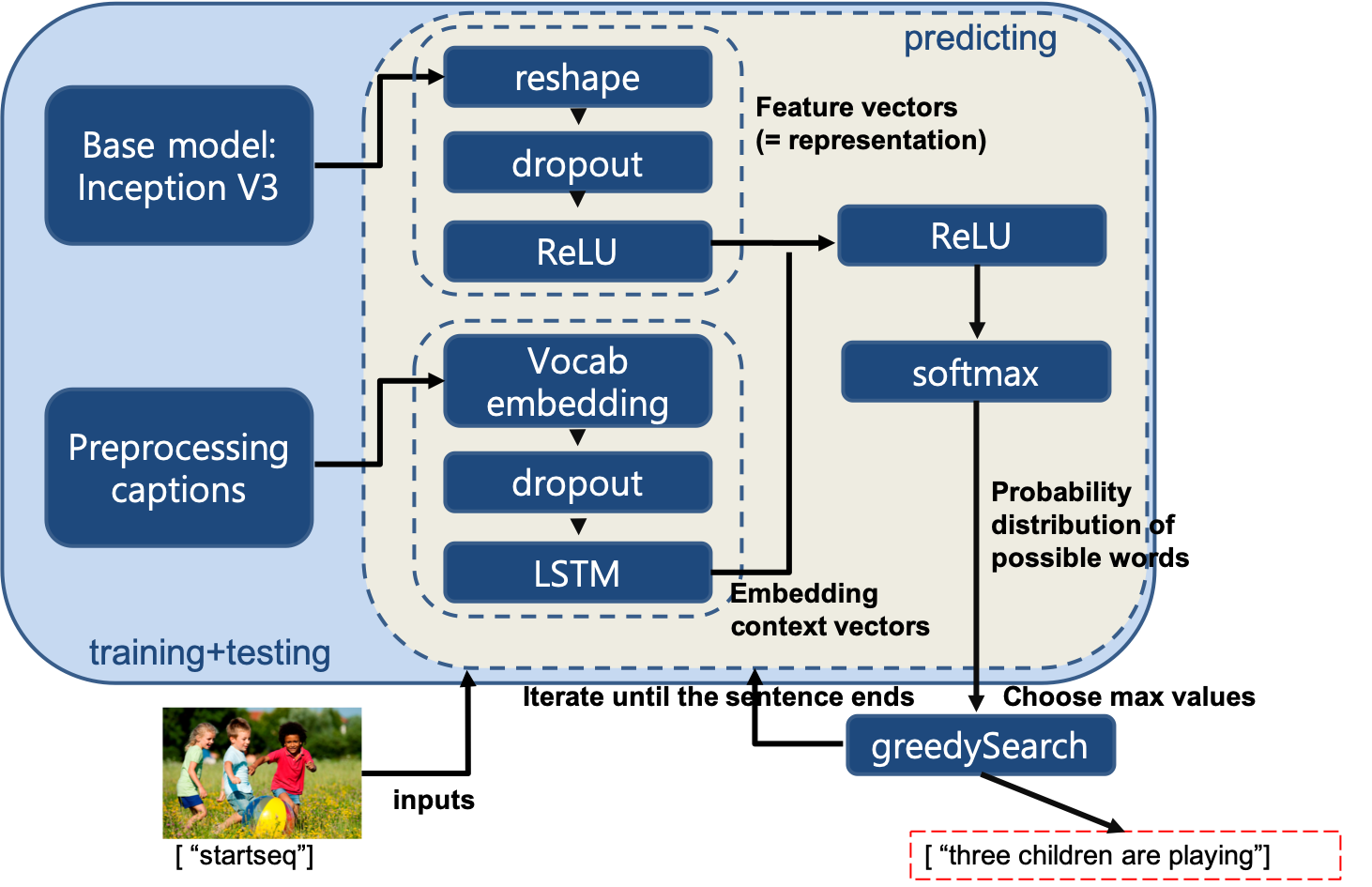
Computational speed not only depends on the feature detection model, but also on the size of the vocabulary—each new word added consumes more time. Just recently scientists have tried to solve the image captioning task by resizing the vocabulary dictionary. Usually the vocabulary size might vary from 10,000 to 40,000 words, while their model relies on 258 words. The decrease is quite sharp—reduced by 39 times if compared to 10,000, but the results are high, with some space for improvements. This means, generating sentences need to embrace a huge dimension without degradation of performance.



Recurrent neural networks (RNNs) have been proposed for learning sequence to sequence tasks. RNNs have widely been used in various sequence learning tasks. However, traditional RNNs suffer from vanishing and exploding gradient problems and cannot adequately handle long-term temporal dependencies which the requirement above may not be satisfied. LSTM networks are a type of RNN that has special units in addition to standard units. LSTM units use a memory cell that can maintain information in memory for long periods of time. In recent years, LSTM based models have dominantly been used in sequence to sequence learning tasks. Another network, Gated Recurrent Unit (GRU) has a similar structure to LSTM but it does not use separate memory cells and uses fewer gates to control the flow of information.

However, LSTMs ignore the underlying hierarchical structure of a sentence. They also require significant storage due to long-term dependencies through a memory cell. In contrast, CNNs can learn the internal hierarchical structure of the sentences and they are faster in processing than LSTMs. Therefore, recently, convolutional architectures are used in other sequence to sequence tasks, e.g., conditional image generation and machine translation. However, it is more common to use CNN as an encoder which understands image and use LSTM as an decoder to produce sentences based on image representation obtained by CNN.

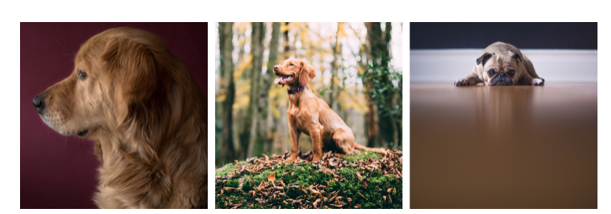
3-4. Proposed deep learning model architecture



3-4-1. Pre-trained Inception model

Inception model is now evolving with updating its version. Basically, for the first version. It wants to deal with following problems.

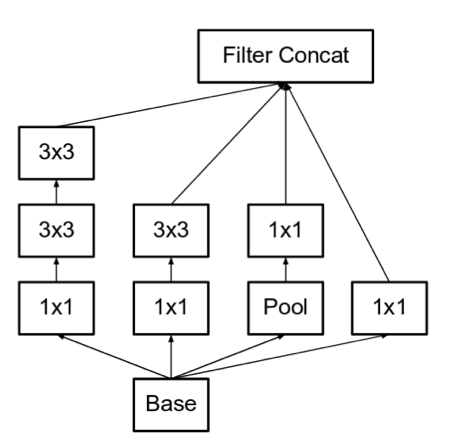
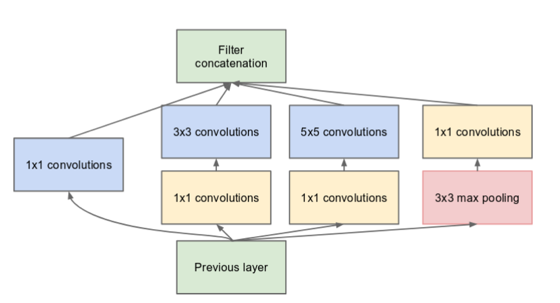
1. Salient parts in the image can have extremely large variation in size. For instance, an image with a dog can be diverse. The area occupied by the dog is different in each image.
2. Because of this huge variation in the location of the information, choosing the **right kernel size** for the convolution operation becomes tough. A **larger kernel** is preferred for information that is distributed more **globally**, and a **smaller kernel** is preferred for information that is distributed more **locally.**
3. **Very deep networks** are prone to **overfitting**. It also hard to pass gradient updates through the entire network.
4. Naively stacking large convolution operations is **computationally expensive**.



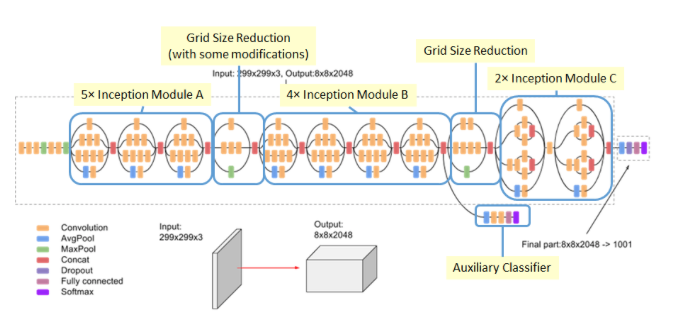
The solution for this is to have filters with multiple sizes operate on the same level.The network essentially would get a bit “wider” rather than “deeper”. The authors designed the inception module to reflect the same.

The below image is the “naive” inception module. It performs convolution an input, with 3 different sizes of filters (1x1, 3x3, 5x5). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module. As stated before, deep neural networks are computationally expensive. To make it cheaper, the authors limit the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions. Though adding an extra operation may seem counterintuitive, 1x1 convolutions are far more cheaper than 5x5 convolutions, and the reduced number of input channels also help. The 1x1 convolution is introduced after the max pooling layer, rather than before.

After then, Inception 2 had been faced with following problems: Reducing representational bottleneck. The intuition was that, neural networks perform better when convolutions didn’t alter the dimensions of the input drastically. Reducing the dimensions too much may cause loss of information, known as a “representational bottleneck. Using smart factorization methods, convolutions can be made more efficient in terms of computational complexity. To handle with this problem, they adopt **to factorize 5x5** convolution **to two 3x3** convolution operations to improve computational speed. Although this may seem counterintuitive, a 5x5 convolution is **2.78 times more expensive** than a 3x3 convolution. So stacking two 3x3 convolutions in fact leads to a boost in performance. This is illustrated in the below image.

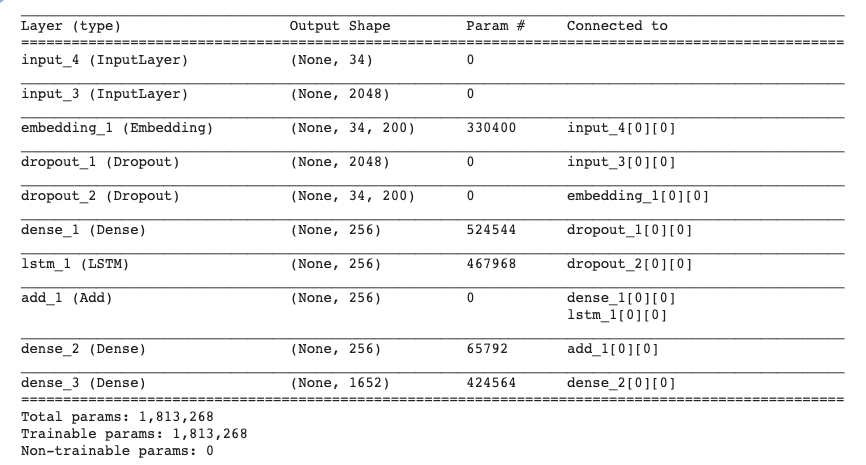


Inception v3, which we adopted, is made that the authors noted that the auxiliary classifiers didn’t contribute much until near the end of the training process, when accuracies were nearing saturation. They argued that they function as regularizes, especially if they have BatchNorm or Dropout operations. Possibilities to improve on the Inception v2 without drastically changing the modules were to be investigated. Thus, Inception Net v3 incorporated all of the above upgrades stated for Inception v2, and in addition used the following: RMSProp Optimizer, Factorized 7x7 convolutions, BatchNorm in the auxillary Classifiers, Label Smoothing (A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class ultimately preventing over fitting).



3-4-2. Additional process

We used pretrained inception model in keras to make image representation vector to catch the feature of the image as an encoder. Then before before feeding them directly to the unlinear function(ReLU), we stack several more steps before combing each result of inception and LSTM. The added model is summarized as below.



For training, we used Flilckr8K contains 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events. The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.

1. Application Service Implementation

Basically, the application service is combination between web service and the trained deep learning model. Therefore the web service is a very important part beside the deep learning model. In order to build a web service, ReactJS and Flask framework are used to develop back end and front end of the targeted web service.

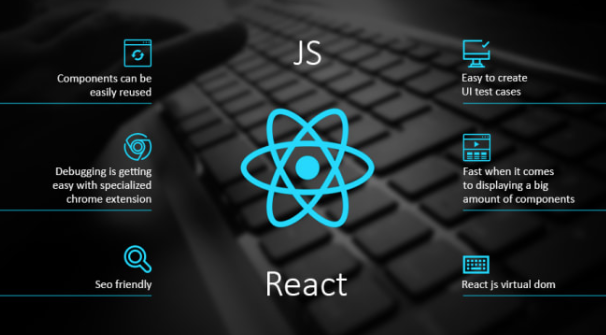
4-1. React JS

React JS is an open-source, component-based Javascript library, which is used to build user interface specifically for single-page applications. It’s one of the most popular Javascript libraries used to build apps front-end right now. Facebook developers created ReactJS in 2011 and used first in the Facebook app, and until today it has a huge community supporting it and lots of resources to learn it. Javascript is a programming language used to create dynamic elements on the website or web app. Almost all modern browsers support Javascript. Entering any webpage, ithas lots of features written in Javascript, like form validation, advanced animations, popups, etc. Javascript is used to build web apps and pages with client-side scripting JS library is a pre-written Javascript code that makes the development of apps easier and faster. It consists of ready components, functions, and patterns, which can be used to accomplish specific programming tasks. In many cases, it also helps to keep our applications more secure and more efficient. Javascript frameworks are used to decrease the amount of coding and reduce the cost of app development, as libraries are mostly open-source and free.

Virtual DOM: If using pure Javascript, the DOM object will re-render every time any change will be done in HTML elements. It’s fine if you have a static website where not too much happens; the performance is safe. But in the case of dynamic web apps that have lots of user interaction elements, it doesn’t work so good. The performance of the application goes down significantly. Creators of ReactJS decided to handlethis issue, and they created a Virtual DOM. When changes are made in the DOM, ReactJS creates a copy, called Virtual DOM. This copy is compared with the normal DOM, and only the element which is different is re-rendered. It takes less computing power, and loading time, that’s why it’s a good improvement

JSX: In ReactJS, JSX is used for creating templates instead of HTML. JSX is a kind ofextension of Javascript, and it allows us to use HTML tags inside the Javascriptcode. It’s used to create templates in ReactJS. ReactJS is a very popular technology is used not only to create your startup ideas, but also huge companies use this technology for huge projects. Currently, there are many popular apps created with ReactJS:

* Facebook
* Instagram
* Netflix
* New York Times
* Dropbox
* Yahoo! Mail

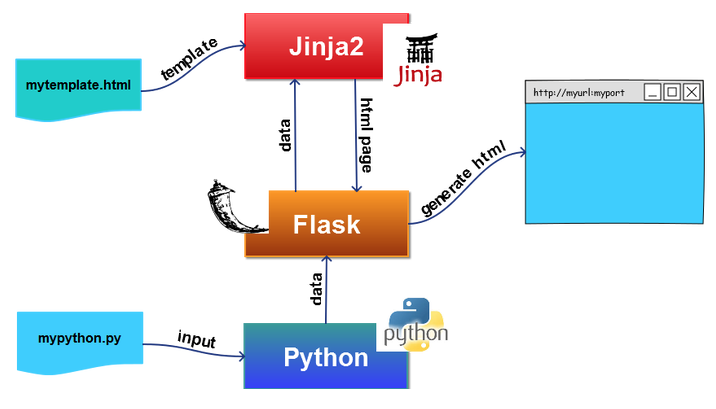


4-2. Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

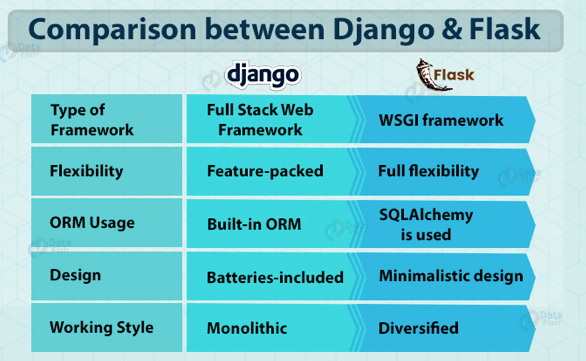
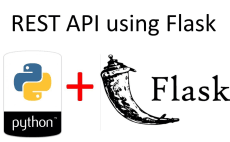
4-2-1. Main components of Flask

1. Werkzeug is a utility library meant for usage with the Python language. Mostly, it is a Web Server Gateway Interface or WSGI app that can create software items for request, response, or utility functions.
2. Jinja is a template engine for Python programming purposes, and it resembles the Django web frameworks templates.



4-2-2. advantages of using Flask

1. Flask’s main advantage is that it has access to a multitude of online resources for documentation purposes.
2. It is one of the most used Python web frameworks, which is why there are available a lot of tutorials or libraries for it.
3. Also, what recommends Flask as the go-to structure is its minimalist approach that does not lose form its power. It is simple to run with either vanilla HTML or bootstrap functionalities.
4. Also, Flask is excellent for creating quick prototypes due to the useful tools included in its package. You can opt for ORM or exact SQL, while all the information related to it is available in the vast array of documentation.
5. Some drawbacks arise from using Flask. First of all, there is no async programming included, while the default setup for designing apps in Flask is quite challenging in terms of reusability and code cleanliness. Besides, a disadvantage of using Flask might be the HTML-oriented structure, which is not meant for creating APIs but still allows the design of APIs.

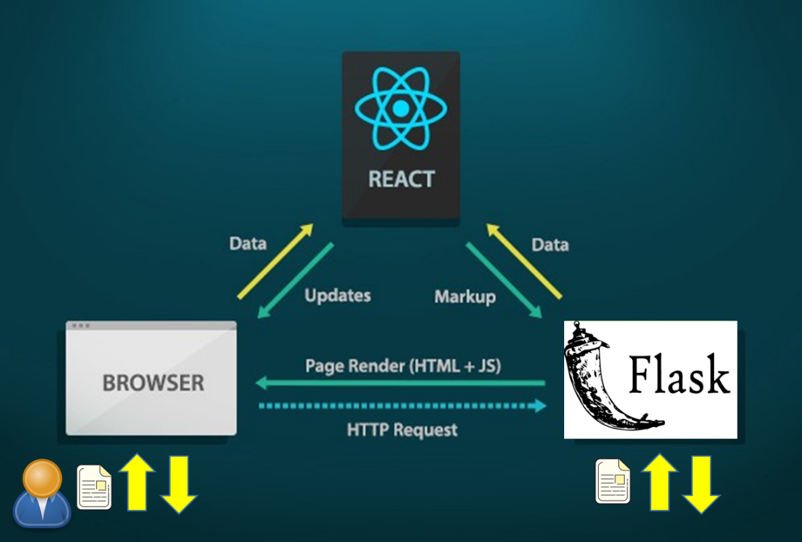
 

There are many frameworks that we can use to develop the backend server such as node.js express. However, in this project, the purpose is to integrate the deep learning model with web service. Therefore, using Flask bring a very significant benefit because it support python as programing language. Using python the deep learning model can be easily embedded inside the web service to provide our targeted application service

4-3. Web service architecture

The combination of Flask back end server and react.js front end can be archived as above figure. Functionally, in this application, the front end provide an interface to user so they can upload the raw file to the server and download processed file (which contain the images with corresponding captions). Technically, the user and back end server interact with each other throw the react.js front end. The React front end communicate with the Flask back end by using the APIs. Base on the web service, the basic concept of this application can be describe as flowing steps:

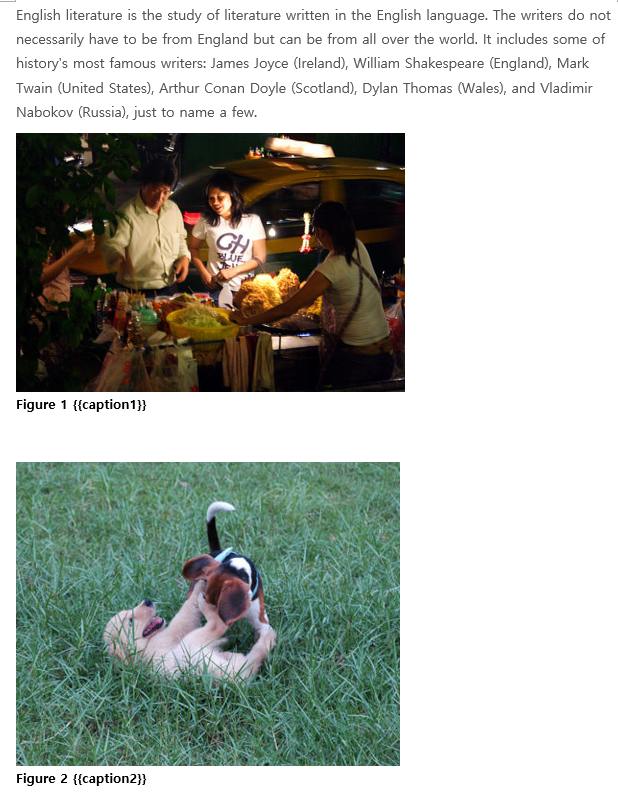
1. Through the interface, the user upload the targeted document, which is .docx or .doc format
2. React.js front end call API to send the file to back end side
3. Flask server save the file and run the deep learning model to process it, the processed file will be saved in a specific folder.
4. When the user request the processed file, a request is sent to the Flask server from React front end. Then the Flask server will return the processed file immediately to the user.



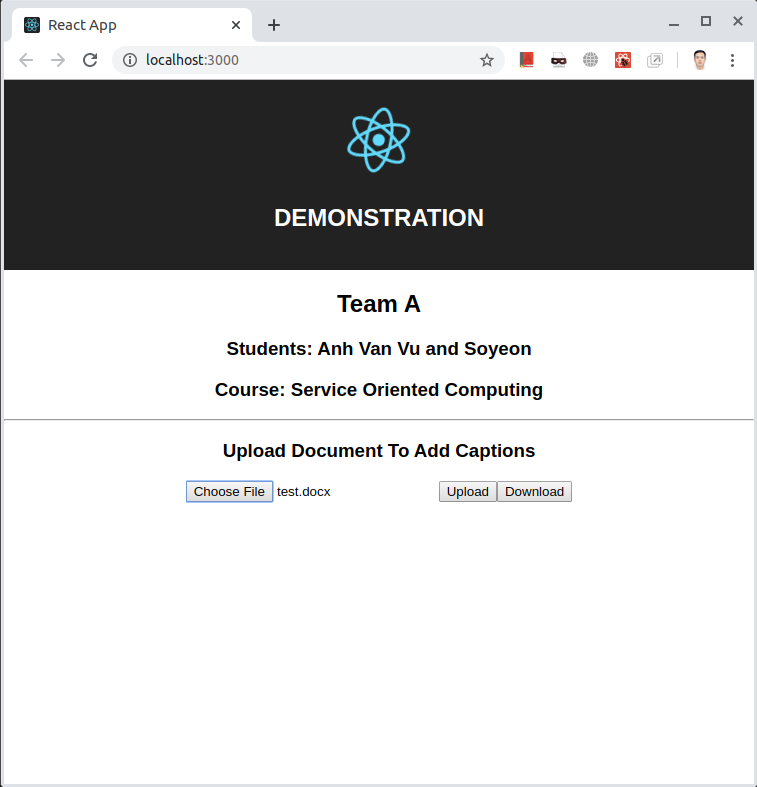
1. Result and Discussion

To see how the application work, we have constructed a test with the following step:

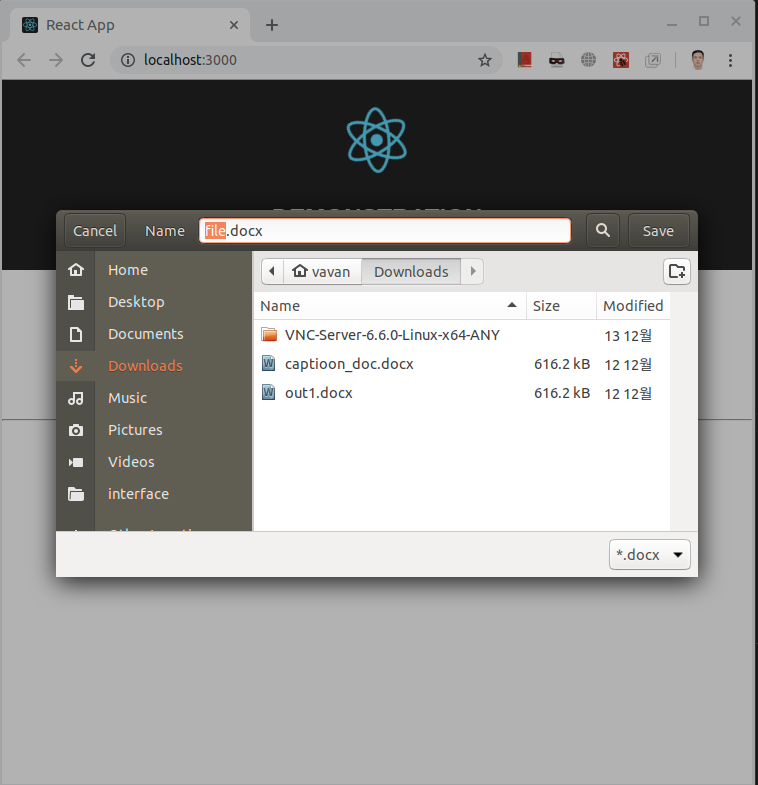
1. Step1: Create a document that contain some image without caption(The created document is attached as appendix 1)



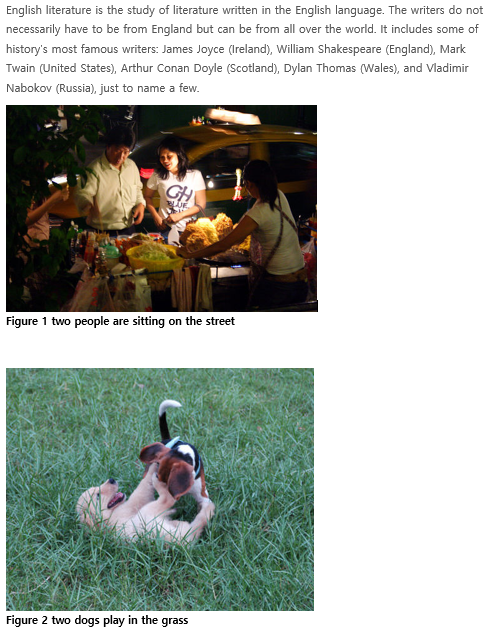
1. Step2: Upload the targeted document to the server throw the web interface to use the created application service



Step3: Download the processed file and see the result.(The proceed file is attached as appendix 2)



Please refer the appendix to compare the original document and the processed document.



As the result getting by application service, all the images in the original file have been captioned. The content of the caption is quite close to the content of the figures. Practically, the content of the captions not only depend on the content of the images but also depend a lot on the text in the document. Therefore, to generate the caption more precisely, we should consider a different deep learning model that use both image and text of the document as input. This is what we think to do improve this result for future work.

1. Conclusion

In the framework of this team based project. We have created an application service, which is combination between service oriented computing and deep learning. The main idea is to insert captions to images in a document, that potentially help document creators save time and so increase their performance. As the results shown in previous session, we have archived a prototype of the targeted application service. We hope that, it can be a great idea for any developer who have enough ability and supports to improve and integrate with Microsoft Word.

Thank you so much for reading our report! Merry Christmas and Happy new year!

Appendix 1:

English literature is the study of literature written in the English language. The writers do not necessarily have to be from England but can be from all over the world. It includes some of history's most famous writers: James Joyce (Ireland), William Shakespeare (England), Mark Twain (United States), Arthur Conan Doyle (Scotland), Dylan Thomas (Wales), and Vladimir Nabokov (Russia), just to name a few.



Figure 1 {{caption1}}



Figure 2 {{caption2}}

**Why Is Studying English Literature Important?**

Okay, so there are about a thousand things for a teenager, or even a 50-year-old adult, to do in today's wired, 500-channel cable television world. We can watch feature films on our phones or hop in a car and drive a hundred miles away in just a couple of hours. That's not how things used to be. People used to read literature for entertainment because even just 50 years ago, there were simply not many readily available entertainment options.

### Middle English Literature (1150-1485)

Some scholars would argue that the **Middle English** period started as early as the 1100s. However, because the English language did not evolve into a dialect we could understand today until about the 12th century, 1150 seems like a better place to start.



Figure 3 {{caption3}}



Figure 4 {{caption4}}



Figure 5 {{caption5}}

Appendix 2:

English literature is the study of literature written in the English language. The writers do not necessarily have to be from England but can be from all over the world. It includes some of history's most famous writers: James Joyce (Ireland), William Shakespeare (England), Mark Twain (United States), Arthur Conan Doyle (Scotland), Dylan Thomas (Wales), and Vladimir Nabokov (Russia), just to name a few.



Figure 1 two people are sitting on the street



Figure 2 two dogs play in the grass

**Why Is Studying English Literature Important?**

Okay, so there are about a thousand things for a teenager, or even a 50-year-old adult, to do in today's wired, 500-channel cable television world. We can watch feature films on our phones or hop in a car and drive a hundred miles away in just a couple of hours. That's not how things used to be. People used to read literature for entertainment because even just 50 years ago, there were simply not many readily available entertainment options.

### Middle English Literature (1150-1485)

Some scholars would argue that the **Middle English** period started as early as the 1100s. However, because the English language did not evolve into a dialect we could understand today until about the 12th century, 1150 seems like a better place to start.



Figure 3 white dog is running through the grass



Figure 4 woman walking down sidewalk street



Figure 5 boy in red shirt is playing soccer ball