UNIVERSITEIT VAN AMSTERDAM

BACHELOR OPLEIDING KUNSTMATIGE INTELLIGENTIE

Real-time Composing of Restaurant Label Classifiers utilising Semantic Word Similarity

Author:
Tony NGUYEN

Bachelor Thesis
Credits: 18 EC

Supervisor:
Dr. Efstratios GAVVES

Co-supervisor:
Noureldien HUSSEIN

Faculty of Science
Science Park 904
1098 XH Amsterdam

June 24, 2016

Abstract

Yelp proposed a problem where insufficient labels of restaurants are predicted by the use of user-submitted photos. Inspired by this problem, an attempt was made for on-the-fly construction of new restaurant label classifiers. This thesis proposes an approach to compose new label classifiers in real-time utilising semantic word similarities. A new classifier can be constructed based on a set of pre-trained concept classifiers and the semantic relations among concepts. The two main contributions in this thesis are firstly, the performance comparison of a classifier trained on original data and one trained on augmented data, and secondly, a semantic search engine capable of classifying new concepts eliminating the need for additional data and training. Various experiments were conducted to improve overall performance as well as the practicality of the classifiers. In addition to the proposed contributions, a comprehensive analysis of the Yelp data set and the various methods for data acquisition for the aforementioned components are present in this thesis.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Efstratios Gavves, for his support and guidance throughout the research. His continued support led me to accomplish the goals of my research.

I would also like to extend my sincere appreciation to my co-supervisors Noureldien Hussein and Kirill Gavrilyuk for their earnest assistance and outspoken advice during my research.

Special thanks to Dennis Koelma for his invaluable lecture and support on DAS4.

Contents

List of Figures

List of Tables

List of Abbreviations

Restaurant **l**abel	**RL**
User-**s**ubmitted **p**hoto	**USP**
Convolutional **N**eural **N**etwork	**CNN**
Fully-**c**onnected	**FC**
Neural **N**etwork	**NN**
Red, **g**reen and **b**lue	**RGB**
Graphical **p**rocessing **u**nit	**GPU**
Rectification non-**l**inearity **u**nit	**ReLU**
Concept **c**lassifier	**CC**
Semantic **s**earch **e**ngine	**SSE**
Part of **s**peech	**POS**
Support **v**ector **m**achine	**SVM**

1 Introduction

1.1 Kaggle

Statistical analysis in the field of machine learning requires a vast amount of data; this is an obstacle which is necessary to overcome. It is difficult for a person or team to start on a real world problem with the a lack of actual data. While a possible solution for this would be to manually aggregate or even hand-make data to work on a particular problem, it has proved to be time consuming and tedious. Therefore, a platform called Kaggle[1] exists where these large data sets are made publicly available along with a proposed problem. This is executed by companies with large bodies of data to attract people who are eager to participate in a challenge. With competitions ranging from *structure identification in images* to *the detection of states in sequences of images* and with leading companies ranging from *Home Depot* to *Expedia*, high-performing individuals or teams can compete in these competitions for knowledge-gain, job opportunities or even money. See Figure 1 for examples of competitions hosted by leading companies.

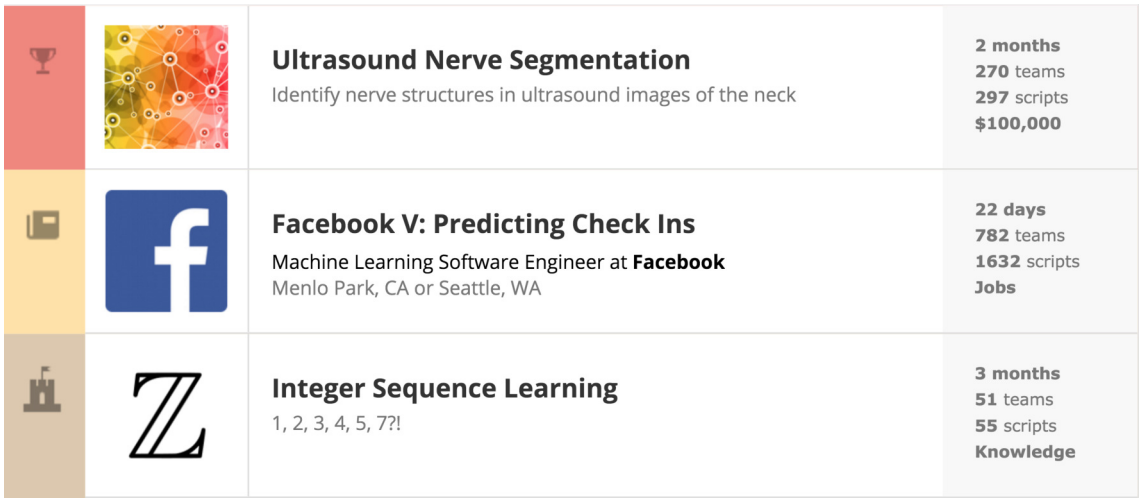

Figure 1: Various Kaggle competitions hosted by leading companies. Rewards from top to bottom are: money, job opportunities and knowledge.

1.2 Yelp

One of the many companies that host competitions on Kaggle is Yelp, which is a crowd-sourced review platform for local businesses, where registered users can submit reviews of these local businesses (See Figure 2 for the user interface of the Yelp mobile application). On the platform registered users can discover new or search familiar local businesses, ranging from restaurants to clubs and from small shops to big-box stores. Reviews submitted can be done in written text, shares, likes, check-ins and even photos which Yelp can subsequently use to make accurate recommendations for users about local businesses based on their preferences. For example,
recommendations generated for restaurants are based on labels assigned by these reviews.

[1] https://www.kaggle.com/ - accessed Fri June 24th 2016

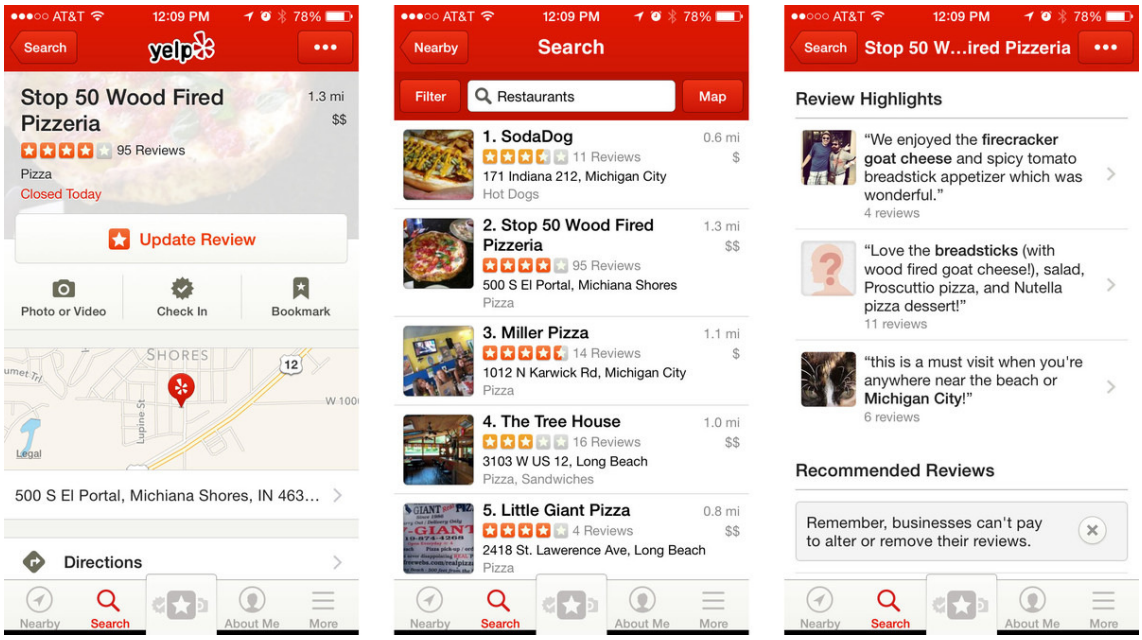

Figure 2: Yelp mobile application user interface. For searching, discovering and reviewing of local businesses.

Reviews can have a significant impact on the reputation of a local business. Negative reviews especially can impinge on the public perception of a business and can be the fault of the local business itself or due to Yelp mistakenly publishing erroneous information. This information often consists of labels that are not intended to be associated with that particular business for which a possible cause could be the lack of information for correctly inferring these labels. Although users are free to submit reviews as they like, with many options such as tagging their photos with suggested labels, this particular process is optional and thus results in incomplete data.

Yelp recently[2] hosted a competition with the proposed problem being that a set of their businesses were missing label; a business was conceived as being from their selection of restaurants and their labels from descriptive tags for that particular restaurant. These restaurant labels (RL's) are what helps Yelp to index their set of restaurants and the users with respect to the choice of where to eat. There are nine distinct RL's (see Table 1) for which a restaurant can be assigned with one or multiple, meaning that the restaurant possesses that characteristic.

Table 1: List of restaurant labels

ID	Restaurant label
0	good_for_lunch
1	good_for_dinner
2	takes_reservations
3	outdoor_seating
4	restaurant_is_expensive
5	has_alcohol
6	has_table_service
7	ambience_is_classy
8	good_for_kids

[2]Start date: Mon 21 Dec 2015 - End date: Tue 12 Apr 2016

To undertake the problem a data set consisting of user-submitted photos (USP's) is provided, a sample of which are displayed in Figure 3. At first glance, it is evident the data set can contain various types of photos often depicting food, receipts, people, outdoor and indoor scenery and menus. The relations between the photos, restaurants and RL's are depicted in Figure 4. To elaborate on the problem, Yelp uses the RL's to quickly answer questions like *'Does your favourite Ethiopian restaurant take reservations?'*, *'Will a first date at that authentic looking bistro break your wallet?'* to *'Is the diner down the street a good call for breakfast?'* and narrows down results to only restaurants that fit the nuanced needs. Currently, RL's are manually selected by Yelp users when they submit reviews, the selection of which is optional, leaving some restaurants un- or only partially-categorised. Therefore, with this competition Yelp challenges people to turn USP's to words.

Figure 3: A sample of the photos contained in the Yelp data set illustrating that there are various types of photos.

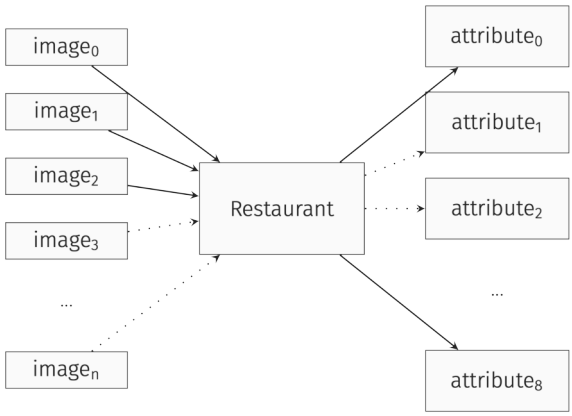

Figure 4: A visualisation of relations amongst data. A restaurant (middle) can have one or multiple restaurant labels (right) and user-submitted photos associated (left) to it.

The remaining part of this thesis will built on what is introduced previously and is structured as follows; firstly what the thesis aims to contribute to the field and its objectives will be stated; secondly, a literature review highlighting the important historical advances leading to the
state-of-the-art technology used in this thesis will be presented. Thirdly, a detailed analysis of the problem proposed by Yelp and on the published data set is discussed. Fourthly, the approach, evaluation methods and results for the individual parts are laid out. Fifthly, a discussion on what was achieved will be included followed by future research suggestions. Lastly, the overall conclusion to the work in this thesis will be presented.

1.3 Objectives

The primary purpose of this thesis is to introduce an approach constructing new RL's in real-time, which will eliminate the need for new data and model training. The secondary goal is to research whether salient concepts can be extracted from the data set or not, which, in turn, is used to enhance classification performance.

This results in the following research questions: *Can concept classifiers contribute to better classification performance?* and *Can new restaurant labels be introduced using concept classifiers?*.

With its sub-sequential parts being:

- What is the performance of a baseline approach where the training and prediction of restaurant labels is performed directly on user-submitted photos?
- What is the performance of a separately built concept classifier based on frequent used words in Yelp reviews?
- Will the performance of restaurant label prediction improve with the additional data from the concept classifier?
- What approach allows for real-time construction of new restaurant labels?

Three major experiments were designed to resolve these questions. All experiments are constructed to be dependent on one another, with the intention of limiting repetitiveness during the implementation phase. Additionally, each experiment will pose as milestones for progress management throughout this research. The objectives of each experiment is described bellow.

1.3.1 Baseline

The purpose of this part is to construct a model that can predict RL's directly from USP's with high performance. For this a naive approached is considered so all complications that the data inhibits can be ignored. A straightforward model will be constructed and will form the baseline. Improvements, in the form of a concept classifier (CC), proposed in this thesis will aim to increase the performance as well as adding a direct and practical use to the data.

1.3.2 Concept classifier

The classification of concepts in images can be seen as the extraction of prominent and more concrete notions out of data. These concepts are context dependent and will add more relevance the process of RL prediction. To illustrate this let us consider the following example; annotating a restaurant with the RL *has_alcohol* it can depend on various aspects. A photo taken at the restaurant of a glass of beer has more relevance to this RL than say, a photo taken of the interior design. By integrating the CC that can classify more concrete concepts such as glasses, tables and lamps, extra information can be extracted from data and will give a context to the photos. Thus, the photo of the glass would have more weight when associated with the RL *has_alcohol* than the photo of the interior design.

1.3.3 Semantic search engine

The aim of better performance is the one of the two goals of this thesis. The most valuable part is an approach to compute new RL classifiers using the separately built CC, which means that a new RL, say *champagne* can be computed in real-time. The approach utilises the semantic similarity of words in a language and will offer great flexibility for composing new RL classifiers. With this newly computed *champagne* classifier the semantic search engine (SSE) can return photos from the Yelp data set that contain anything related to *champagne*. Not only can this occur without the need of re-training the classifier, more importantly, it eliminates the need of new data.

2 Background

Human task automation has been a practice for years and to this day it is still the case. By 1957 computers were in common use in many research institutes and commercial establishments. While these machines were originally only devoted to algebraic, numerical and geometric computations, later, the symbol manipulation capability of computers became recognised, leading to the use of computers in so-called business data processing when alphanumeric processing became routine. Along with the introduction of alphanumeric data the problem of data scarcity emerged since a vast quantity is required for accurate business data processing. This resulted in developing machinery for character recognition [1]. With this, it occurred to researchers that a general purpose computer could be used to simulate the many character recognition logics that were made in hardware [2]. Although this has been considered a crude step towards the field of image processing and image pattern recognition, it launched the field to the state it is today.

2.1 Convolutional Neural Networks

Major improvements in the field of pattern recognition have been made since the first introduction of an automated approach. Machine learning techniques, with regard to Neural Networks (NN's), have played an increasingly important role for pattern recognition applications. It is arguable that these learning techniques have been a crucial factor in the success of pattern recognition applications, due to the vast amount of learning techniques available. Using multilayer NN's with backpropagation researchers at AT&T constructed a technique for synthesising a complex decision surface which can classify high-dimensional patterns. This technique is used for handwritten digit recognition and requires minimal preprocessing of data. Convolutional Neural Networks (CNN's), introduced by Fukushima [3], designed for this task, outperform techniques that require a hand crafted feature extraction and trainable classifier modules.
Additionally, due to the structure of the designed CNN, this classification technique offers robustness for variance in rotation, scale and transposition. The algorithm, LeNet-5 [4], consists of 7 layers with trainable parameters and takes input of 32×32 pixels. See Figure 5 for the detailed visualisation of the LeNet-5 architecture. Input images are at a maximum of 28×28 pixels to which borders are added to account for end-points that are located at corners of an images, which contain valuable information. Two sets of convolutional and subsampling layers to extract high activation points and reduce dimensionality are followed by two fully-connected (FC) layers which squash the activations to produce an output. After training, the results of the classifier are outstanding when compared to other techniques discussed in the paper. With an error rate of 0.95% after 10 passes and 0.35% after 19 passes, all without the occurrence of over-training.

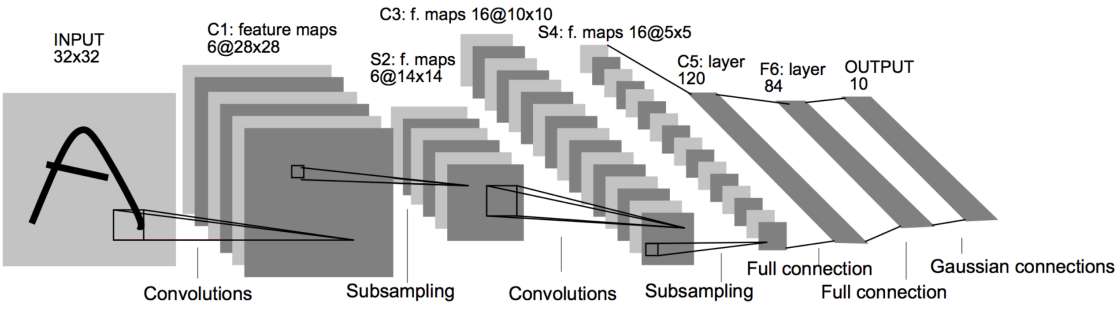

Figure 5: Architecture of LeNet-5, a Convolutional Neural Network, here for digit recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

2.2 ImageNet

The increasing size and complexity of NN's results in an increased necessity for data. Likewise, the availability of images on the internet is also increasing. While the explosion of image data on the internet can potentially contribute to the large demand of data, it does not do so in its current form. For training NN's images need to be annotated with labels, which is usually done manually and is a laborious process. Fortunately, researchers from Princeton University have created a database called ImageNet [5] which has an extensive collection of high-resolution images along with manually annotated labels. As of 2010 ImageNet has over 15 million high-resolution labelled images belonging to roughly 22,000 categories.

Annual competitions called ImageNet Large-Scale Visual Recognition Challenge are held to benchmark existing and new techniques within the image recognition field. The data set used in the competition is a subset of ImageNet with roughly 1000 images in each of 1000 categories. In 2010, a submission completed by the University of Toronto with a deep CNN [6], named AlexNet, ranked top-1 and top-5 with error rates of 37.5% and 17.0% respectively, which performed considerably better than the previous state-of-the-art. AlexNet consists of five convolutional layers, some of which are followed by max-pooling layers, and three FC layers with a final 1000-way soft-max layer (see Figure 6 for its architecture). Images from the data set are of size 224×224 pixels and in red, green and blue (RGB) colour channels. Thus, the input is $224 \times 224 \times 3$ resulting in a more complex problem when compared to 32×32 for digit recognition. To cope with this, CNN's connect local neighbourhoods of the input to nodes in the next layer; with this, local information is preserved and the complexity is reduced. By utilising multiple graphical processing units (GPU's) the time for training increased significantly because NN's and this problem is highly parallelisable. The researchers used two GPU's and their fine-tuning resulted in a decrease of error rate for top-1 and top-5 error rates by 1.7% and 1.2% respectively.

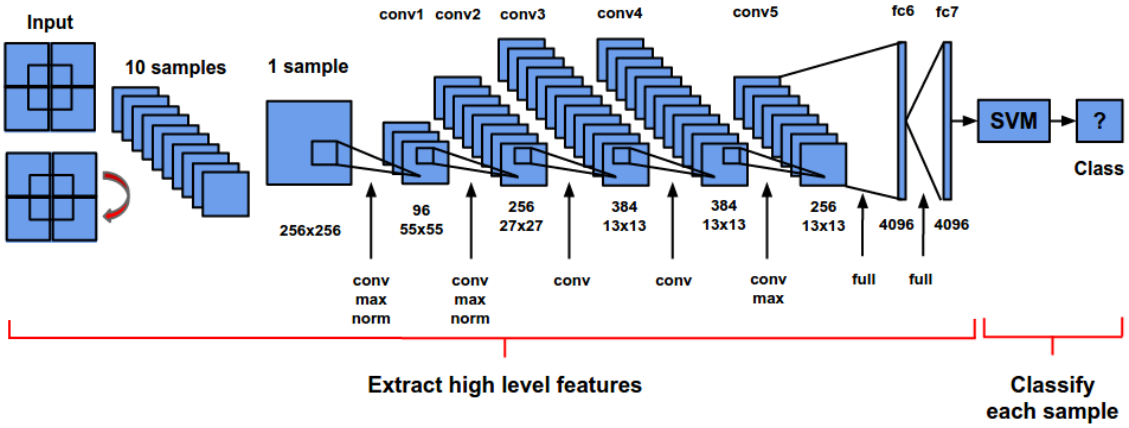

Figure 6: Architecture of AlexNet including a support vector machine classifier, here for image classification[7].

2.3 Feature extraction

For a model to be trained on images directly, large amounts of computational power is required since the images are reasonably large. An image of size 224×224 pixels translates to a vector of 50,176 data points and is impracticable for machine learning, thus, another representation is recommended. Even if all data were to be used to train a model, noise present in the images will affect the performance of the classifier and thereby yielding lower accuracies. Thus, the aim is to feed the most valuable data for the classifier to train on. This process of obtaining the most valuable data is termed feature extraction, which is the process of selecting the most prominent characteristics out of data. These characteristics are features of an image and contains data with minimal noise.

The layers of a CNN propagate activations from lower level layers to higher and more abstract layers. For a CNN to learn concepts, its lower level layers show behaviour similar to the detection of elemental features, such as lines and edges, which are propagated forward where higher layers can construct more abstract features, including multiple lines and edges forming shapes (see Figure 7 for a visualisation of the propagates activations and their internal representations of data).

Activations on higher abstraction levels offer more concrete and prominent features that are present in an image since they combine multiple lower level features to one integrated abstract concept. In VGG-16 these higher abstraction layers are the FC layers of a CNN, with which all direct underlying neuron activations are coupled (see Figure 7 for a visual of the FC layers and their connections with underlying layers). The activation values of a FC layer are more expressive and compact compared to the original input data. By using either FC layer 6 or 7 a feature vector of merely 4096 dimensions can be constructed, which is a reduction of complexity by a magnitude of 12.

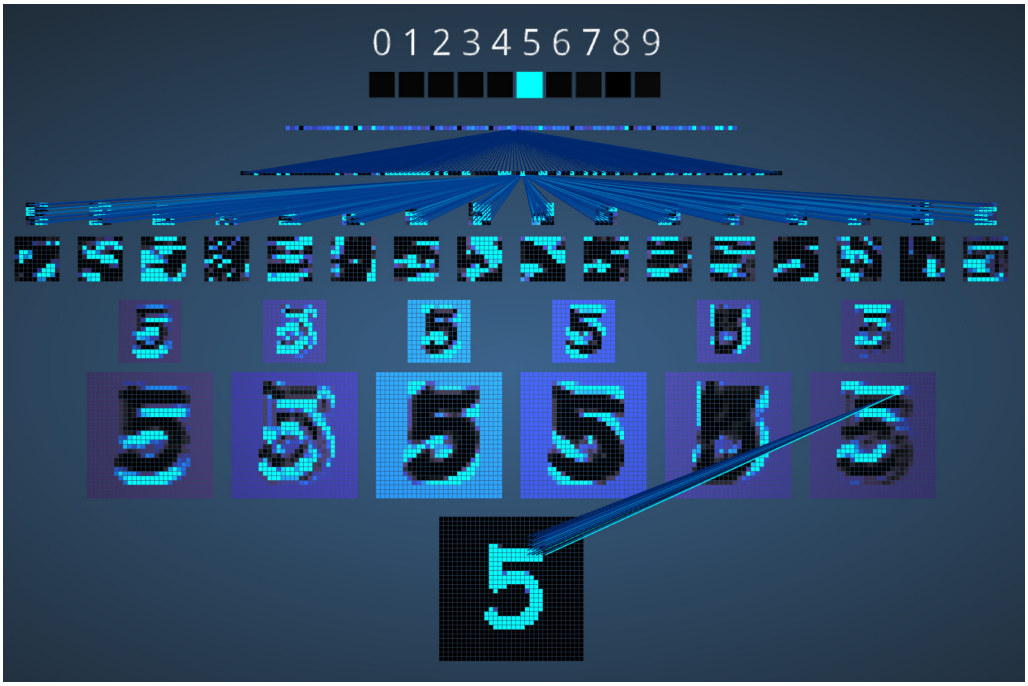

Figure 7: A visual representation of a convolutional neural network for digit classification[4]. The bottom layer are convolution layers for convolutions and subsampling, while the top layers are fully-connected layers.

2.4 Very deep convolutional networks

With tools like Caffe, the performance of image classification systems has increased to new heights. However, the increase was not only achieved by these toolkits, advancements in CNN's in general have also contributed greatly. By constructing a deeper CNN, which means more hidden layers, more invariance is achieved as well as greater performance for large-scale image recognition. Pushing the layer depth to 16-19 layers the researchers at University of Oxford achieved first and second places in the localisation and classification tracks respectively during ImageNet Challenge 2014 [8]. During development, optimal parameters were found and kept fixed, while the depth was steadily increased by adding more convolutional layers. This was feasible due to the use of very small (3×3) convolutional filters in all layers. Input to the CNN was fixed at $224 \times 224 \times 3$, for RGB images, with only mean RGB value subtraction involved in preprocessing. Following some of the concolutional layers, there are a total of five max-pooling layers which perform max-pooling over neighbourhoods of 2×2 with stride 2. The layers of convolution and max-pooling are followed by three FC layers, first and second of which have 4096 channels each, the third having 1000 channels. Lastly, a layer of soft-max is added for mapping the output vector of probabilities for

[4]http://scs.ryerson.ca/ aharley/vis/conv/flat.html - accessed Fri June 24th 2016

binary classification. All hidden layers use rectification non-linearity unit (ReLU) as activation functions. While performing great during the challenge, it was shown that the representation also generalises well to other data sets, in addition to obtaining state-of-the-art results.

Numerous papers publish that by changing the hyperparameters, utilising smaller receptive window sizes [9], smaller strides [10], and greater depths of layers of a CCN the performance will increase. A different approach was proposed [11], where RGB images together with depth data is passed into a CNN resulting in 4 channels (3 for colour and 1 for depth) on which the CNN can train. Better results are achieved with this approach due to its equal treatment of the depth channel and the colour channels. Although this approach incorporates new data to enhance the performance of the classifier, there are cases imaginable where the data set is limited to RGB images only and collecting sensory data is impossible, due to the nature of data acquisition.

2.5 Concept classification

The possession of extra sensory data is always a good addition to the original data set, but often extra data is not present or cannot be gathered. In these cases data augmentation can be a possible solution which is the process of augmenting the original data set to generate new data for image augmentation operations where cropping, transposing and change illumination can be applied. Applying this process would mean that data is introduced that possibly is not present in the real world since they are hand-crafted. Furthermore, results can be less generalised to real world cases which leads to the proposal of another approach, namely the extraction of local features from the original data set often referred to as a CC. Instead of only feeding the CNN raw pixel data, local features like objects can be included as well [12]. These concepts can range from objects such as a table, house, person to a glass, adding extra dimensions of features for the classifier to learn. These concepts are not a result of data augmentation since they are extracted from the original data and thus can contribute to better performance while being representative of real world cases.

2.6 Word2Vec

As with language, the different concepts from the CC have underlying relations, namely their semantic word similarity. Word2Vec introduced in 2013, is an approach for the representation of words in vector space [13, 14], which will enable computations on words and is saliently demonstrated by their powerful example: vector(”King”) - vector(”Man”) + vector(”Woman”) resulting in a vector that is closest to the vector representation of the word Queen.

While the architecture of Word2Vec is a Skip-Gram model (see Figure 8) and is very similar to any other NN, the difference is that there is only a single hidden layer. This architecture is to essentially train a model that can automatically compress an input vector and decompress it back to the original in the output layer, the goal of which is to learn the weights of the hidden layer. When this principle is applied to the domain of linguistics, on words, a representation can be built that results in the added ability to apply computations on words.

The construction of such a model is achieved by training it to produce, for a given word, what the probability of every other word is within a small window. To elaborate, given the word *United* the probability of the words *States* and *Kingdom* appearing nearby will be higher when compared to the probability of the words *penguin* and *camel*. These probabilities will be computed for every word in the vocabulary. Data required for this training task appears in the form of word pairs and will enable the network to learn the frequencies of said word pairs. Because it is evident that NN’s do not accept text strings as input, words are represented as one-hot vectors, with lengths equal to the size of the vocabulary. A word is indicated by a one at its position in the vector while all other positions are set to zero. The output of the NN is a vector containing, the probability that each word would appear near the input word for every word in the vocabulary. This output contains essentially the aforementioned word similarity values.

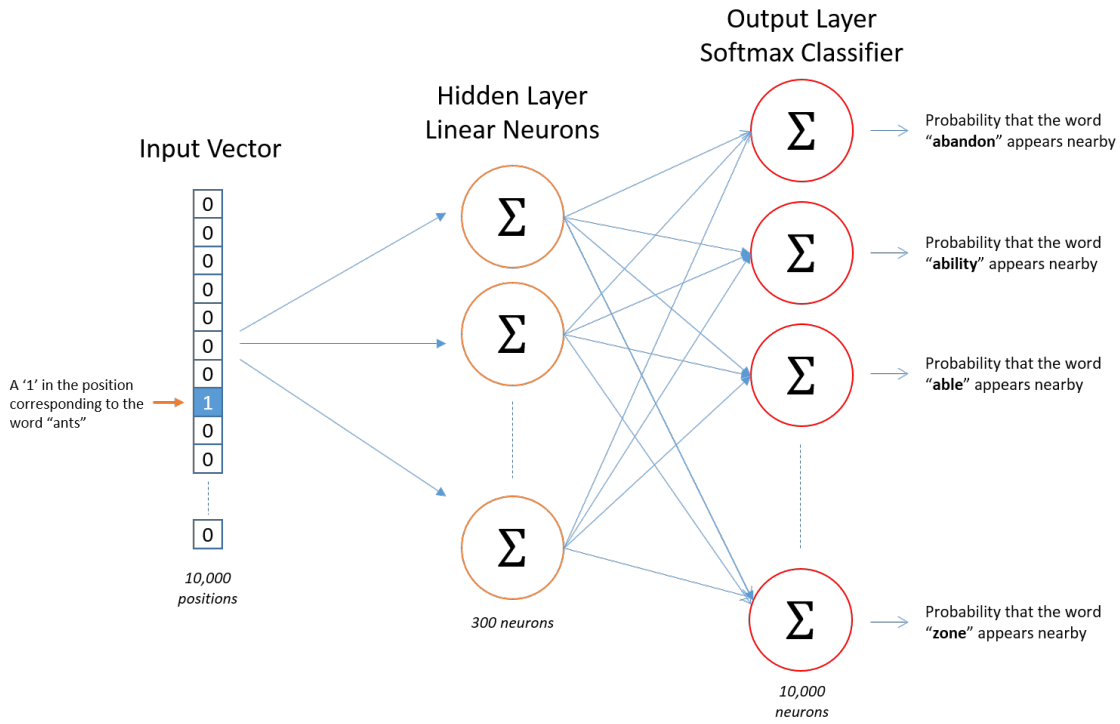

Figure 8: A visual representation of the Skip-Gram model used in Word2Vec[6]. The input is a one-hot vector representation of a word and the NN will output a vector with at each word position the probability of that word appearing nearby the input word.

[6]http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/ - accessed Fri June 24th 2016

3 Approach

3.1 Implementation

3.1.1 Caffe

Building upon the availability of ImageNet, researchers from Berkeley University published a clean and modifiable framework for state-of-the-art deep learning algorithms, which helps prototyping and deploying deep learning systems effortlessly and rapidly. This framework is called Caffe [15], and allows for experimentation and seamless switching from development to production. Caffe adheres to industry and internet-scale media needs by utilising NVIDIA CUDA[7] technology and large-scale distributed clusters [16]. The framework is capable of processing over 40 million images a day on a single K40 or Titan GPU. While programmed natively in C++, Caffe also has wrappers for MATLAB and Python. The latter is significant for research due to the fact that a vast amount of deep learning toolkits are also available in Python. The mix of easy development and deployment with high parallelism makes Caffe a suitable framework for developing deep NN's for image recognition.

The extraction of feature from images will be completed with the Caffe framework utilising VGG-16. Activation values of FC 6 is chosen for the construction of feature vectors as it offers a good combination of abstraction from non FC layers and complexity resulting from being directly connected to underlying layers.

3.1.2 Model

Labels assigned to images can be lengths of one to nine; considering there are nine unique RL's means the classifier should predict a set of target labels. This process is named multiclass-multilabel classification and means that there are more than two classes and that the predicted properties of a data-point are not mutually exclusive. Depicted in Table 3 are restaurants having one or more unique RL's; the classifier should also have this behaviour and output a set of target labels.

A frequently used framework for machine learning is Scikit learn[8]. It offers a wide range of well written and documented algorithms, and by utilising the power of parallelism and optimised code, numerous models can be constructed and trained rapidly.

The framework accepts data with samples as rows and features as columns and thus data is required to be formatted to fit that constraint. For the model to be trained the data matrix has to be formatted, which results in a matrix of dimensions 175,000 $\times$ 4096. This allows the algorithm to iterate the samples individually and possess all features in one vector (see Table 9 for a representation of the data matrix). For the framework to recognise that predictions are multiclass and multilabelled, the ground truths have to be binarized. Binarization is the process of encoding the vector of ground truths into a vector holding binary values, with an one and a zero for a positive and negative label indicator respectively. For example, a ground truth vector with labels [1, 2, 5, 8] will be encoded to a binary vector with values [1, 1, 0, 0, 1, 0, 0, 1].

To test this, a straight forward approach is adopted, which entails that the data samples being feature vectors of centre-cropped images and binarization of ground truths.

Figure 9: Matrix X containing feature vectors of data samples.

$$\mathrm{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,4096} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ x_{175000,1} & x_{175000,2} & \cdots & x_{175000,4096} \end{pmatrix}$$

[7] CUDA offers parallel computation capabilities by harnessing GPU's, see more at http://www.nvidia.com/object/cuda_home_new.html - accessed Fri June 24th 2016

[8] http://scikit-learn.org/ - accessed Fri June 24th 2016

3.2 Baseline

To have a general pipeline for the data to come in and predictions to come out along with performance measures, a baseline setup is constructed consisting of various modules and these also used in the construction of the CC and SSE. In chronological order of use are oversampling, aforementioned feature extraction, data splitting, training, testing and performance measurement. Oversampling of images and splitting of data in training and testing sets are both covered in the previous section. The primary modules of the baseline are discussed in more detail here.

The complete data set provided by Yelp consists of two separate smaller image data sets, one with labels and the other without named train set and test set respectively. The train set is made available for model training while the test set is used for benchmarking the final approach. Since the goal is not to compete in the Yelp competition on Kaggle, the test set will be omitted and not considered throughout this thesis.

A brief analysis on the data set reveals that there are 234,842 image with RGB colour channels. Also provided are two files, one containing photo ID's with the respective restaurant ID's and the other one restaurant ID's with their respective RL's. See Tables 2 and 3 for a representation of the first and second files respectively.

Table 2: Example of photo ID's with their respective restaurant ID's

Photo Id	Restaurant Id
49820	234
12304	234
28922	234
89043	527
81232	391
62390	391
23959	201
38695	201
54903	201

Table 3: Restaurant ID's with their respective restaurant labels

Restaurant Id	Restaurant labels
234	0, 1, 2, 5
527	2, 8
391	0, 1, 2, 3, 4, 5, 6, 7, 8
201	8
421	2, 3, 6
871	1
323	0, 1, 2, 6, 7, 8
555	0, 1, 2, 5, 7, 8
938	3, 6

The the baseline pipeline is visualised in Figure 10.

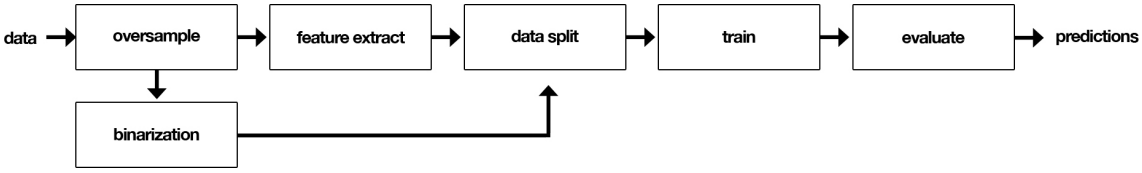

Figure 10: A schematic visualisation of the baseline pipeline.

3.2.1 Data splitting

Training and testing will be performed on smaller subsets of the data set; this is due to limited time and computational resources. A split is made at the ratio of approximately 70/30, meaning the train set will contain 70% of the data set whereas the test set will contain the remaining 30% of data. This reduces the number of training data to 175,000 samples and the number of samples for testing to 59,842 with confirmation that the data split is splitting classes correctly among both sets, which means that all classes are within each set.

3.2.2 Preprocessing

Images in the data vary in sizes and thus are not of compatible format for feature extraction. For feature vectors to be extracted using the Caffe framework with the VGG-16 model[8], images need to be of size 224×244 pixels for which additional preprocessing is required. The shortest side of the image will be resized to 256 pixels, while maintaining the aspect ratio of the image. This is followed by a centre-crop taken of size 224×244 pixel, a preprocessing task that is default for VGG-16 and results in a more efficient trained model. Negligible parts of the data are lost after the centre-crop, due to important parts of an image is usually being middle centred.

3.2.3 Model construction

Training the baseline classifier was carried out with two different algorithms, namely a NN and with a linear support vector machine (SVM). The motivation for the choice of a NN is due to the size of the data set; essentially it means that a large NN can learn patterns with much greater precision. So the hypothesis is that a NN with a larger hidden layer size will perform better. The choice for a linear SVM comes from the fact that the data complexity is reduced remarkably and thus opens up opportunity for a linear classifier to tackle this multiclass-multilabel problem. For both, different hyperparameters were tested to obtain most optimal results. Although optimisation was of no priority, the difference hyperparameter choices was based on trail and error during the learning period of using the framework.

3.2.4 Noise reduction

The approach of RL prediction directly from USP's adds noise to the equation. To illustrate this, take restaurant r with RL's a_1, a_2 and a_3 and USP's divided into sets P_1, P_2, P_3 and P_4. The photo sets P_1, P_2 and P_3 are associated with RL's a_1, a_2 and a_3 respectively meaning all photos in P_1 have some feature that leads to the restaurant having RL a_1, P_2 to a_2 and P_3 to a_3. While by training on these three photos sets and RL's the prediction will be 100% accurate, the actual performance decreases due to the fourth photo set containing photos without any relevance to any of the three RL's. With the introduction of a separate CC this noise will be at minimum, this is due to extra data from the CC added to the original data. Dominant concepts extracted from sets P_1, P_2 and P_3 will now contribute more strongly to their respective RL's, whereas concepts from P_4 will have minimal contribution to the prediction and thus result in improved performance.

3.3 Concept classifier

Due to the approach of predicting labels directly from images, minimisation of noise is the main purpose of the CC. By extracting salient concepts that are located within images, context and relevance can be provided for model construction. Concepts which are possibly present in the data set are for example *chairs*, *tables*, *glasses* and *types of food*. Incorporated in the training phase, aforementioned concepts constructs relation between the found concepts and RL's. To elaborate, restaurants with the label *has_alcohol* can have images that contains some container holding liquids. If this container is a glass, a CC with the ability to classify glasses will be able to extract the concept *glass* and increase the weight of relevance between the concept *glass* and the RL *has_alcohol*.

The hypothesis is that, with the addition of the CC, the baseline performance improves. The assumption that with richer data the classifier can learn more abstract patterns and can in turn predict with higher probabilities. The hypothesis is shown mathematically in Equation 1.

$$\sum X_b \cdot W_b = y_b \text{ where } X_b \text{ is the feature matrix,} \quad W_b \text{ the baseline model and } y_b \text{ the ground truth matrix.} \tag{1a}$$

$$\sum X_c \cdot W_c = y_c \text{ where } X_c \text{ is the feature matrix,} \quad W_c \text{ the concept classifier model and } y_c \text{ the ground truth matrix.} \tag{1b}$$

$$\sum X_b \cdot W_c = y_{bc} \text{ where } X_b \text{ is the feature matrix,} \quad W_c \text{ the concept classifier model and } y_{bc} \text{ the predicted labels.} \tag{1c}$$

$$X_b + y_e = X_e \text{ where } X_b \text{ is the feature matrix,} \quad y_e \text{ the predicted labels, } X_e \text{ the new feature matrix and "+" is the process of concatenation.} \tag{1d}$$

$$\sum X_e \cdot W_e = y_b \text{ where } X_e \text{ is the new feature matrix,} \quad W_e \text{ the enhanced baseline model and } y_b \text{ the ground truth matrix.} \tag{1e}$$

$$\text{Hypothesis: } e(W_e) > e(W_b) \text{ where, } e(x) \text{ measures the F1 score performance of model } x. \tag{1f}$$

In this part of the experiment there are also different modules that need to be constructed, namely, data and corresponding label acquisition for model construction and the production of concepts for the CC to learn. After the data has been acquired it will be passed on to the baseline pipeline to train the model and for evaluation.

3.3.1 Concept selection

The selection of the most suitable concept type is determined by four criteria. Firstly, in preparation for the SSE part of this thesis, the concepts should have underlying similarities which can be used to derive new RL's. Secondly, the occurrences of the concepts have to be frequently used so more statistics can be extracted from them. Thirdly, the concepts are required to be visually predictable; this implies that the images should clearly describe the concept. Lastly, there should be visual similarities between concept images. In addition, the model has greater convergence in the train phase when samples have more similarities amongst them.

The first, second and third criteria dictates that the most suitable type of concept should be words, because they have lingual semantic similarities which enables it to be computed on. Another advantage about the use of words is that they can be scrapped and extracted from actual Yelp reviews. Occurrences of particular words will be high due to the sheer volume of user-submitted reviews, also meaning the probability of the

presence of highly visual predictable words is higher. Similarity of image criteria is of no concern since there is a large set of concepts which will in turn balance out the less predictable concepts.

3.3.2 Acquisition of concepts

To acquire a set of words along with their frequency counts, user-submitted reviews were required to be scrapped from Yelp. This part demands two main steps, namely the collection of unique restaurant ID's and scrape user-submitted reviews of those restaurants. Yelp has unique identifiers for all their restaurants which makes scraping reviews difficult to automate.

There are times the user of Yelp does not know what local restaurant to choose, which inspired the makers of *random restaurant generator*[9] to make a website which can suggest local restaurants at random. By utilising this website, a list of restaurants and their corresponding restaurant ID's can be made. The generator website has one mandatory input parameter for it to work; it must be supplied with an US address. For this a list of all US states is made and of each state the restaurant ID's will be scrapped. The automation of this process was done by a straightforward Python script, which visits the website in intervals, queries based on location for a new restaurant, extracts the web-link and from the web-link the restaurant ID's.

Reviews are scrapped directly from the Yelp website page source, due to the fact that Yelp does not support for anyone to scrape reviews of their site. By injecting the restaurant ID to a web-link, the automated scrapper can access the restaurant's page along with their reviews, which are divided amongst multiple page tabs and a list is made to index all those tabs. Then each index is visited by the scrapper and all user reviews are directly extracted from the page source.

3.3.3 Clean data

Utilising user-submitted reviews directly for the extraction of concepts can invoke problems, considering the fact the reviews are not of any standard format. Users can write reviews using incorrect grammar, sentence structures and emoticons. To tackle this problem only valuable words are considered as candidates for concepts. A word is considered as valuable if it meets criterion three (Section 3.3.1) - that concepts are required to be visually predictable. While words such as *the*, *a* and *and* are frequently used, which conforms with criterion two, they are not visually predictable and thus considered invaluable.

To eliminate these type of words, part of speech (POS) tagging is applied to all words. POS tagging is the process of assigning POS tags, including noun, verb and adjective to words. This is done using the Natural Language Toolkit[10] which offers a powerful POS tagging tool for POS tagging large sets of text. An example of POS tagging can seen in the sentence:

- *At eight o'clock on Thursday morning ...*

with as result:

- *[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), ...]*

Proposed here is that only nouns and adjectives will be selected for concept candidates because these two conform with the four criteria.

To bring computations to a minimum, all words will firstly be frequency counted to minimise the amount of words the POS tagger has to tag. Due to time and computational constraints, only the top 500 words will be used for the CC. See Table 4 for a selection of the filtered top 30 most frequent words along with their frequency counts.

[9] http://www.restaurantgenerator.com/ - accessed Fri June 24th 2016
[10] http://www.nltk.org/ - accessed Fri June 24th 2016

Table 4: Top 30 most frequent words with their frequencies.

food	214,185	order	40,771	salad	34,974
place	183,949	people	40,189	lunch	33,893
service	97,657	dinner	40,140	beer	33,097
menu	73,656	sauce	39,087	sandwich	30,985
restaurant	71,427	night	38,863	bread	29,333
cheese	50,037	burger	38,352	drinks	27,642
bar	48,306	fries	37,538	pizza	26,003
table	47,088	staff	36,921	day	25,645
chicken	41,497	dish	36,829	meat	25,138
meal	40,998	pork	36,003	lobster	24,908

3.3.4 Acquisition of data

The next step after the extraction, filtration, and selection of the top frequent words, is to scrap Google for images. For each concept the top 100 images will be scrapped as taking only the first 100 images ensures that these images are the most relevant to the concept, since Google sorts image results based on relevance. A python script will query each of the 500 concept words on Google Images and download them locally.

3.3.5 Baseline pipeline

With the data in place, the same pipeline as in the baseline can be used. This entails the first part preprocess images for their centre-crops and feature-extracting their most significant features. Secondly the part to train the model, predict unseen samples and measure the overall F1-score performance is outlined.

3.4 Enhancement

Combining the baseline approach with the CC, the extracted data can be enhanced with predictions from the CC. As hypothesised in Equation 1, the aim is to achieve better performance when there is more data for the classifier to learn.

The result of the CC, by passing the feature extracted data of the baseline through it, is a binarized vector of the length 500. This vector is then concatenated to the end of the original feature vector of that particular sample, the outcome of which is a new feature vector of total lengths 4596. Following the same approach as before, the data will be used for model construction, the model to predict RL's and the predicted RL's for performance measurement.

3.5 Semantic search engine

Producing new RL's is not an effortless task; it involves appropriate data with correct labels and extensive time to train. With the approach for the SSE, this process can be carried out in real-time. New RL classifiers can be constructed with the aid of the CC, which is done by utilising the intrinsic semantic meaning of the concept words. The computation of the new RL classifier weights vector N is determined with Equation 2 and each individual part is described below.

$$N_j = \sum_{i=1}^{500} a_j^i \cdot v_j \tag{2a}$$

where, N_j is the new classifier, a_j restaurant labels,
a^i concepts and v_j concept classifier weights

as matrix multiplication notation

$$N = n \cdot V \tag{2b}$$

where, N is the new classifier weight vector, n the new word similarity vector
and V the classifier weight matrix

This equation requires two parameters, namely vector n and matrix V. Vector n contains word similarity values, which are cosine distance values dictating how closely two terms are related to each other (see Table 5 for words with the closest cosine distance to the word *france*). While computing the word similarity between the new RL word and the 500 concept words, this process essentially decomposes the word in terms of the concept words. Consider the analogy where a car is decomposed to its most basic parts such as wheels, frame and engine. A representation can be constructed where each part is expressed as a percentage weight of the car, so $part_{weight}$ = [wheels 25%, frame 60%, engine 15%]. Whenever $part_{weight}$ is given, it refers back to the car. This is similar to the construction of n; the word is decomposed into 500 parts, with each in possession of their own weights. For new RL words containing multiple words such as *greek_food*, word similarity values will be computed for each individual word and the average of the resulting vectors will be taken as n.

The new RL word *champagne* decomposed with its values in vector $n_{champagne}$ of size 1×500:

$$n_{champagne} = [0.46634, 0.05034, 0.20102, 0.30889, 0.35129, ...]$$

Table 5: Examples of similar words to *france* and the corresponding cosine distances.

Word	Word similarity
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

The CC consists of multiple separate classifiers, 500 to be exact, which operate together with the outcome of the best performing classifier being selected. Each separate classifier has its own set of weights and one bias value that is used to compute the decision boundary. Matrix V is made up of those classifier weights, where the classifiers are sorted in rows with their weights in columns. The dimension of matrix V is 500×4096 (see Figure 11).

Figure 11: Matrix V containing concept classifier weights.

$$\text{V} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,4096} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ w_{500,1} & w_{500,2} & \cdots & w_{500,4096} \end{pmatrix}$$

The result of performing Equation 2 is a vector N of size 1×4096 containing weights for a linear classifier, which can be inserted to an existing classifier and have it predict the newly introduced RL. The mathematical representation of the insertion process of new classifier weights is in Equation 3.

A linear classifier defined by $X \cdot w = y$ (3a)
where, data set $X = [x_1, x_2, x_3, ...]$,
classifier weights $w = [\alpha_1, \alpha_2, \alpha_3, ...]$
and predicted probabilities y

insertion of new classifiers weights vector N

A linear classifier defined by $X \cdot N = y'$ (3b)
where, data set $X = [x_1, x_2, x_3, ...]$,
classifier weights $N = [\alpha'_1, \alpha'_2, \alpha'_3, ...]$
and predicted probabilities y'

4 Results

Domain and data analysis is crucial to understanding the current problem-state and gives guidance to what approaches are favourable in order to solve the problems. First, analytics on the domains are presented, including Yelp and Google and secondly what insights are extracted from the data set. Finally this is followed by results from the experiments and their corresponding analysis.

4.1 Yelp reviews

To build a CC based on the most frequent used words in user-submitted reviews, a method was required to scrape those reviews. While Yelp does offer an interface for querying of reviews, it is limited to 25,000 requests per day. It involves multiple requests to obtain a single review, since Yelp puts hard limits on users who use the interface deliberately for data aggregation purposes, meaning another third-party approach was favourable to bypass Yelp's limits. For this, numerous open sourced methods written in Python were available. The method found most suitable offered extraction of reviews by means of the website page source, which offers great flexibility and adaptation on occasions where Yelp changed their page representation in the source.

The review extraction script was executed for one whole day, to be certain that a sufficient set of unique restaurant ID's was scrapped. In total, 960 unique restaurant ID's were collected and will be used for review extraction. For 960 restaurants this results in 308,424 individual reviews with a total of 35,058,394 words.

4.2 Google images

The acquisition of images from Google required multiple different approaches for it to work. Google also sets limits on users who intentionally use their platform for data collection, meaning time-outs will be given to users when they are assumed to be collecting a large body of data. For this the same approach was used in the collection of images, by means of extraction of the image links and downloading them directly from the source. To avoid the time-out, a process would extract the maximum number of links at intervals and put them into a list, while another process would download the images from the same list resulting in halting period being efficiently utilised. An additional note on the download of images is that only the top 100 images were taken. This is due to the ranking system of Google, where images that appear lower in the search have lower relevance to the concept. See Figure 12, where the images are sorted left for most relevance and right for least relevance.

The total number of images downloaded is 45,420; this number is lower than the theoretical 50,000 images due to some images in the top 100 not being able to be downloaded.

Figure 12: Images from Google sorted on relevance, left for most relevance and right for least relevance.

4.3 Data set

With the data set provided by Yelp a couple of insights were extracted, namely the analysis of the image dimensions, the number of unique restaurants, the distribution of restaurant labels and the distribution of image for each restaurant. Insights gleaned from these contributed to more thoughtful decisions during experiments.

Due to the standardisation of USP dimensions, no images were larger than 500 $\times$ 500 pixels as shown in Figure 13. This was beneficial since all images were resized to 256 pixels, implying that larger dimensions would shrink by a higher factor and would result in blurriness and loss of information. Additionally, the data set contains a significantly smaller set of images below 256 pixels, meaning that only a fraction of the data set is actually up-scaled to 256 pixels and thus added noise is kept at a minimum.

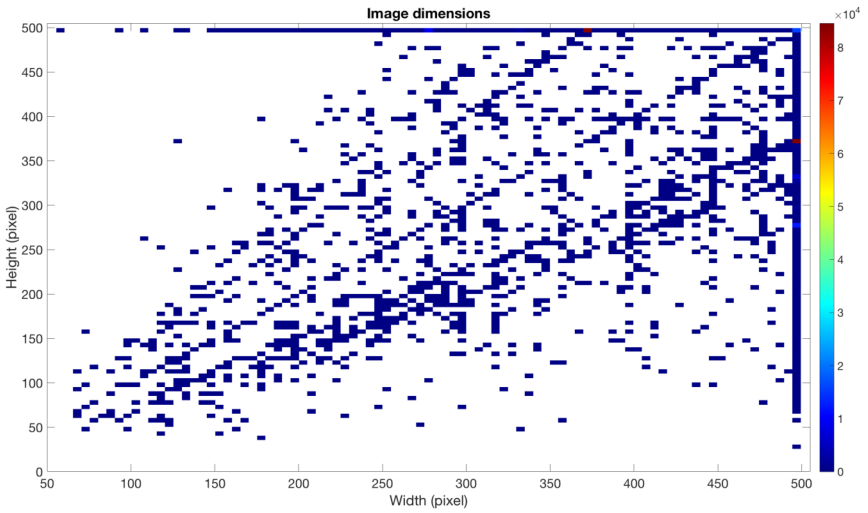

Figure 13: A scatter plot of image widths against respective image heights, including colour densities indicating where more data points are located.

The data set contains roughly 234 thousand individual images, which could be evident in either of the following two scenarios. The first scenario is that all images are divided equally amongst the restaurants whereas the second scenario is where some restaurants have marginally more images than others. An analysis of this in Figure 14 demonstrates that the distribution of images reflects the latter scenario. There are 2000 unique restaurants with an unevenly distributed number of images. To put it into perspective, only a small fraction of the restaurants contain half of all images.

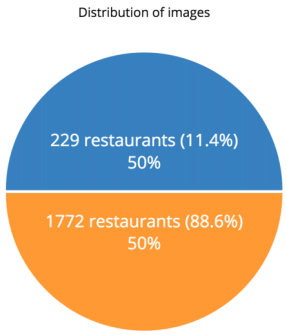

Figure 14: A chart visualising the distribution of images amongst restaurants. A small fraction of restaurants possesses half of the images.

In addition, the distribution of RA's amongst restaurants is also uneven as shown in Figure 15. With over a quarter of restaurants with six RA's, resulting in a modest bias for the prediction of RA's.

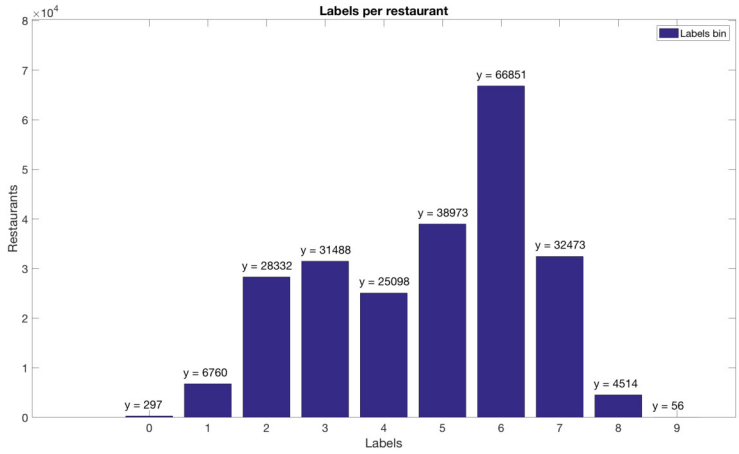

Figure 15: A histogram visualising the distribution of restaurant labels amongst restaurants. There is a bias towards restaurants with six restaurant labels.

4.4 Performance measure

Measurement of performance is enacted in terms of the mean F1-score, also known as example-based F-measure, which is commonly used in multilabel information retrieval. The F1 score, measures accuracy using the statistics precision p and recall r, the former being the ratio of true positives (tp) to all predicted positives (tp + fp) and the latter being the ratio of true positives to all actual positives (tp + fn) (see Equation 4 for mathematical formulation of the F1 score and Figure 16 for a diagram visualising these concepts).

The F1 metric weights precision and recall equally, and an algorithm with good performance will maximise both precision and recall simultaneously. Additionally, it calculates the average for each instance to distribute the metrics evenly. Thus, moderately good performance on both will be favoured over extremely good performance on one and poor performance on the other.

$$F1 = 2 \times \frac{p \times r}{p + r} \text{ where } p = \frac{tp}{tp + fp}, r = \frac{tp}{tp + fn} \tag{4}$$

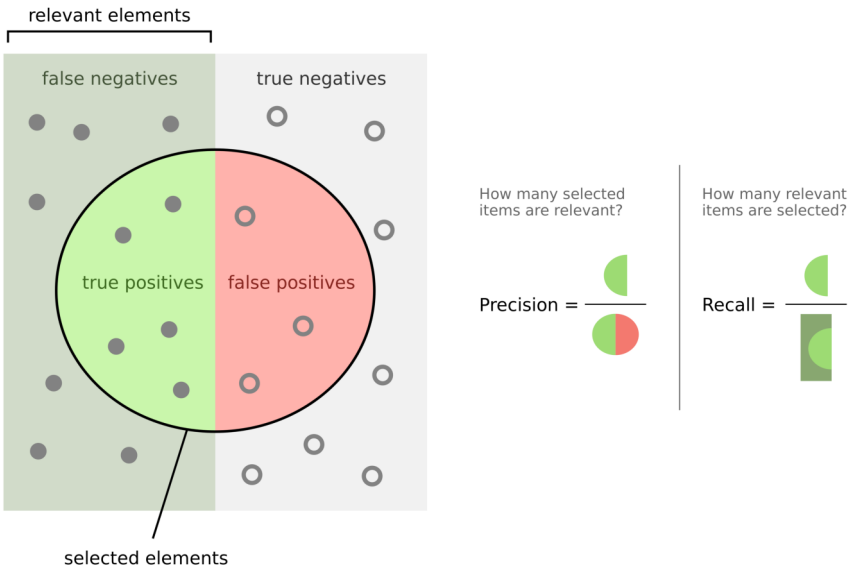

Figure 16: A diagram visualising the four different fractions of instances of data retrieval[11].

[11] https://en.wikipedia.org/wiki/Precision_and_recall - accessed Fri June 24th 2016

4.5 Baseline

Various modules from the baseline pipeline produce output that is used in the adjacent modules, some of which have abstract results that are difficult to interpret. There are two main intermediate results that will be visualised in order to gain more intuition on them; Firstly there is the process of oversampling images and selecting of the centre-crop and, secondly, a graph which offers more insight in the abstract representation of feature vectors.

To oversample an image is to augment the image and save those augmentations which results in more data in the case of a scarcity of data. However, the purpose of this for this thesis is not to generate more data, on the contrary, it is meant to reduce the dimensionality of the data and make it suitable for feature extraction, which is enacted to reduce the complexity of the data even more. The original steps to oversample an image are as follows: resize the image to the appropriate dimension, crop the four corners and centre to the desired size and repeat the crops for the mirrored image (illustrated in Figure 17). However, for the experiments in this thesis only the unmirrored centre-crops will be used.

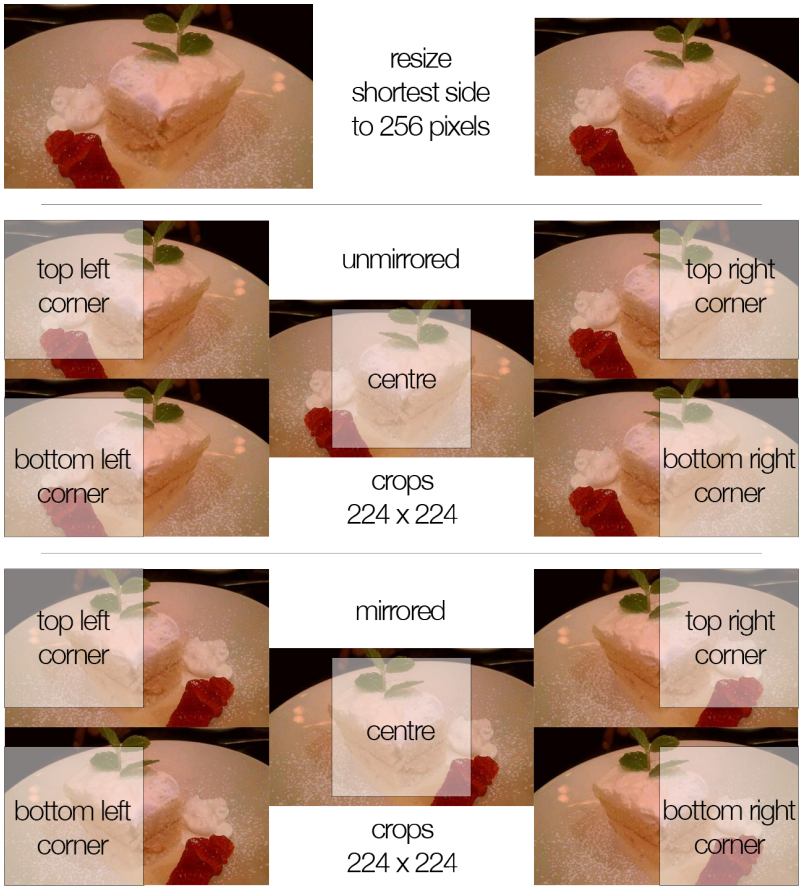

Figure 17: The processes involved in oversampling an image in sequence from top to bottom with steps: resizing the image to the appropriate dimension, cropping the four corners, centring to the desired size and repeating the crops for the mirrored image.

Extraction of features from images entails that only the most prominent characteristics of an image will be used. These features are considered most prominent by VGG-16, exemplified by its model trained on thousands of classes and millions of images. The activation values from FC layer 6 are composed to form a feature vector of a particular image. Output from a neuron in FC layer 6 is a product of the rectified linear unit activation function, meaning there is no upper bound for values but a fixed lower bound at 0.0. There are a total of 4096 neurons in FC layer 6, resulting in there being 4096 individual activation values. Each image is passed through the CNN and a 4096-D vector is outputted (see Figure 18 for a visualisation of such a feature vector). For more intuition, a value of the feature vector represents a firing of a neuron in the NN for which greater firing potentials contribute to higher activation values in a feature vector. Neurons that are tuned to a specific characteristic in an image will fire more intensely when that characteristic is present and

more calmly when it is absent. This firing pattern of the set of neurons is what defines an image; different images produce different firing patterns and thus different feature vectors.

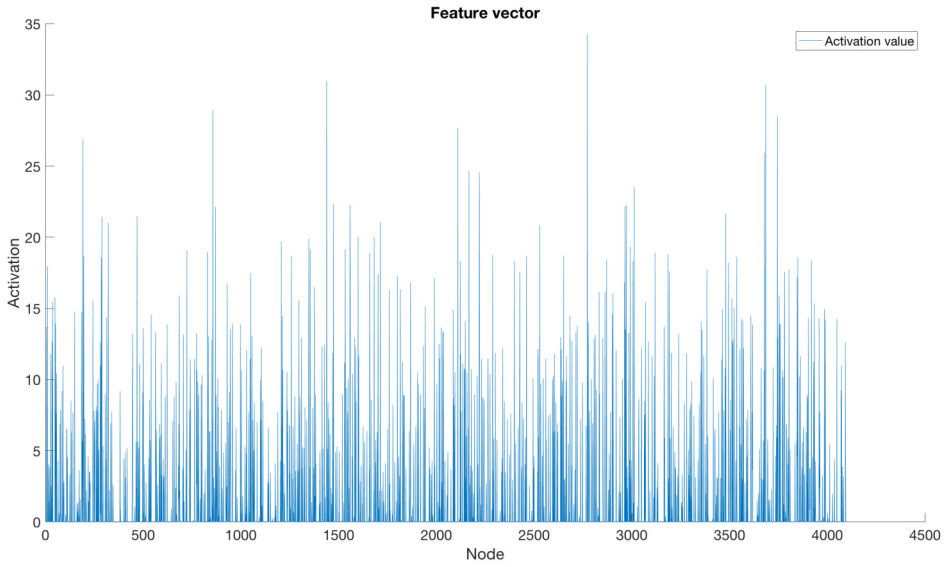

Figure 18: A visualisation of a feature vector containing neuron activations extracted using VGG-16.

The results are remarkably well, with the best performing model for NN's being NN_5 and linear SVM's SVM_8 in Tables 6 and 7 respectively.

Table 6: Neural network F1 score results with different hyperparameters.

	Hidden layer sizes	Activation	Algorithm	Alpha	F1 score
NN_1	10, 10	tanh	adam	1e-06	0.745
NN_2	10, 10	tanh	sgd	1e-06	0.749
NN_3	800, 500	tanh	l-bfgs	1e-03	0.762
NN_4	800, 500	tanh	adam	1e-03	0.766
NN_5	**800, 800**	**tanh**	**adam**	**1e-06**	**0.770** •
NN_6	300, 300	tanh	sgd	1e-06	0.750
NN_7	300, 500	tanh	adam	1e-03	0.751
NN_8	500, 300	tanh	adam	1e-06	0.767

Table 7: Linear SVM F1 score results with different hyperparameters.

	Iterations	Loss	Alpha	F1 score
SVM_1	100	log	1e-03	0.710
SVM_2	100	modified_huber	1e-03	0.672
SVM_3	100	log	1e-03	0.702
SVM_4	100	modified_huber	1e-03	0.714
SVM_5	1000	log	1e-04	0.711
SVM_6	1000	modified_huber	1e-04	0.662
SVM_7	1000	log	1e-04	0.693
SVM_8	**1000**	**modified_huber**	**1e-04**	**0.723** •

The hypothesis for NN's to perform better is proven with the fifth NN, its performance eclipses all other results without any compromises in terms of training time. Based on the results the classifier selected for the baseline will be NN_5.

4.6 Concept classifier

Performance for the classifier is reasonable, even though the data set contains insufficient samples and a large variety of classes. For the construction of the model two algorithms were considered: NN's and linear SVM. Although the NN will not be used for the CC, the reason it was not selected is due to its non-linearity. Weights of the classifier cannot be modified or switched without making it unusable. It is thus essential for the CC to deliver linear weights so these can be used to compute new classifiers. The results in Table 8 show that for this scenario the same NN does not outperform the other classification algorithm, which is understandable as the CC data set is a factor of four smaller compared to the baseline data set. An additional NN was trained with the ReLU activation function since it was proven to perform better compared to tanh [6], although the observed improvement was insignificant.

Table 8: Scores of the neural network and linear SVM classifier with different hyperparameters.

	Iterations	Loss	Alpha	F1 score
SVM_1	**100**	**hinge**	**1e-03**	**0.372** •
SVM_2	100	log	1e-03	0.366
SVM_3	100	modified_huber	1e-03	0.366

	Hidden layer sizes	Activation	Algorithm	Alpha	F1 score
NN_5	800, 800	tanh	adam	1e-06	0.321
NN_6	800, 800	relu	adam	1e-06	0.322

To manually see how the predictions look, three concepts are arbitrarily chosen for display. The images are tagged with their ground truths to visualise whether they are positive or negative predictions. Figure 19 is an example where the large majority is predicted correctly, whereas Figures 20 shows more wrongly predicted instances. Although the third and last set of predictions in Figure 21 shows that all predictions are wrong and thus will produce unacceptable results, by human analysis, it is clear that a few do actually are associated with the concept *ambience*.

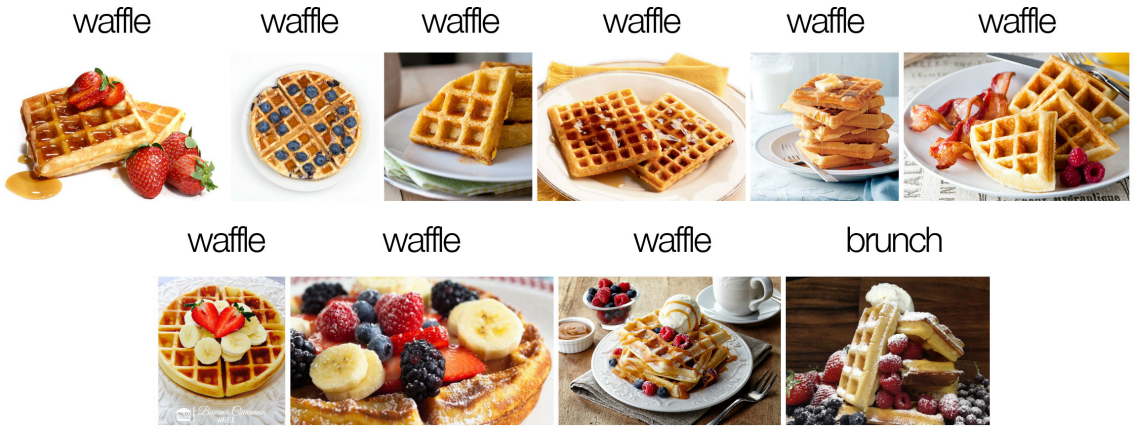

Figure 19: Top 10 predictions for the concept *waffle*. The predicted images are tagged with their ground truths.

Figure 20: Top 10 predictions for the concept *peppers*. The predicted images are tagged with their ground truths.

Figure 21: Top 10 predictions for the concept *ambience*. The predicted images are tagged with their ground truths.

4.7 Enhancement

By incorporating extra salient information extracted with the CC, the goal was to improve the overall performance of the classifier. This however was *not* the result. At best the F1 score performance was not affected and in some experiments the performance even decreased.

Enhancement was achieved by concatenating the predictions from the CC directly to the feature vectors used in the baseline approach. Two approaches were examined, one where the binarized output from the CC was used and the other the probabilities values. The difference between the two is that the binarized output consists of ones and zeros, which means that the classifier will assign no relevance to a zero label and full relevance where there is a one, whereas, with the probabilities, the output is any real value between one and zero and thus still has some relevance assigned to low probability labels. With either one of the approaches the new enhanced feature vector is of length 4596, that is length 4096 of the original and 500 additionally from the CC.

A comparison of the two methods of representing the predictions from the CC inside the original data set showed not noticeable differences. At most, the F1 score was affected by 0.005 and thus adds no benefit to the selection of either one. With this concluded, Tables 9 and 10 shows results utilising the approach of concatenating the binary predictions to the original data.

Table 9: Neural network F1 score results with different hyperparameters trained on the enhanced data set.

	Hidden layer sizes	Activation	Algorithm	Alpha	F1 score
NN_1	10, 10	tanh	adam	1e-06	0.700
NN_2	10, 10	tanh	sgd	1e-06	0.701
NN_3	800, 500	tanh	l-bfgs	1e-03	0.745
NN_4	800, 500	tanh	adam	1e-03	0.755
NN_5	**800, 800**	**tanh**	**adam**	**1e-06**	**0.769** •
NN_6	300, 300	tanh	sgd	1e-06	0.706
NN_7	300, 500	tanh	adam	1e-03	0.711
NN_8	500, 300	tanh	adam	1e-06	0.718

Table 10: Linear SVM F1 score results with different hyperparameters trained on the enhanced data set.

	Iterations	Loss	Alpha	F1 score
SVM_1	100	log	1e-03	0.567
SVM_2	100	modified_huber	1e-03	0.532
SVM_3	100	log	1e-03	0.534
SVM_4	100	modified_huber	1e-03	0.518
SVM_5	1000	log	1e-04	0.658
SVM_6	1000	modified_huber	1e-04	0.642
SVM_7	1000	log	1e-04	0.676
SVM_8	**1000**	**modified_huber**	**1e-04**	**0.680** •

4.8 Semantic search engine

The construction of the SSE required more time and effort because of the inflexibility of the Scikit learn framework. This results in that the SSE is still in an experimental phase, where these results were hard to produce. Furthermore, due to time constraints, no performance measurements were carried out on the SSE, which had the effect of there being no concrete metrics are available for judgement on the performance of the newly constructed classifier. In spite of this, it is possible to select the top predictions from a new classifier and collect the corresponding images. Therefore when utilising human analysis, results seem to be reasonably good. The top 10 predictions usually contain five to seven correct images. With some new RL's performing better due to their relatively higher similarity values, where other are show more falsely classified images. A well performing case is with the word *champagne* in Figure 22, it has seven correctly classified images. Lesser performing cases were of words *crab* and *sushi*, shown in Figures 23 and 24 respectively.

Figure 22: Results on the new restaurant label *champagne* from the semantic search engine.

Figure 23: Results on the new restaurant label *crab* from the semantic search engine.

Figure 24: Results on the new restaurant label *sushi* from the semantic search engine.

The process of constructing a new classifier is done within seconds as all the necessary components are built and stored beforehand. In terms of the experimental design of the SSE, the user is only required to input the desired new RL and everything is done automatically, which meanwhile converts the new RL to a vector of word similarities, combining the newly similarity vector with the CC weights, and load the weights to a classifier. In three computational undemanding steps a new classifier is constructed, without the need of data or any training.

5 Discussion

With the experiments finished and the results presented, the point for a reflection of what has been executed its significance it reached. Starting with the main two research questions: *Can concept classifiers contribute to better classification performance?* and *Can new restaurant labels be composed using concept classifiers?*.

To conclude whether additional data will lead to an improvement of classification performance, a base line was constructed. Measuring the most naive approach to RL classification, the final result yields a F1 score of 0.71, meaning there is a harmonic balance between the risks and certainty of predictions. The baseline approach is a crucial part of the thesis, due to the modules it requires. Since those same modules are also used in later experiments, doing so increases the use of efficient and general purpose code. Following this, is the construction of the CC to extract salient information from the original data, with a F1 score of 0.372 which is acceptable considering the size of the data set, and with this answer the first research question. Hypothesised was that concepts containing context will reduce the added noise and improve the overall performance of classification. This however was not the case. Due to the low performance of the CC, concepts extracted from the original data were insignificant and did not improve classification performance.

For future research constructing a CC of more concepts is suggested to aid in the broader
decomposition of new RL's, in addition to the utilisation of a larger data set. A data set of a factor 10 larger is recommended, which is comparable to the data set size Yelp issued.

The second research question, of whether a separate built CC can be used to compute new RL's or not has also been verified. Experiments and results show this process is possible, with the added major benefit that such a classifier can be computed in real-time, bypassing the conventional approach of data acquisition and model construction entirely. However, it has been shown to be restricted by some limitations; words of new RL suggestions are required to be present in the vocabulary of Word2Vec and it is recommended that the words are frequently used. This will enable the computation of the word vector and indicates the similarity between the new RL and the existing concepts. Words with high frequency are preferred as well, since frequently used words contain more semantic relations with other words and result in better similarities measurements.

There are additional improvements to the current approach; firstly, the process of handling new RL's and secondly, a framework for word similarity with a more extensive vocabulary could be incorporated. The domain of restaurants and food is partially covered by Word2Vec and there are numerous types of food and brands that are not present in the vocabulary. By utilising lemmatization of words, which is a process for removing the commoner morphological and
inflectional endings from words, the probability of the word residing in the vocabulary is increased. Another consideration would be to find the parent or child of the word within the word hierarchy.

A further improvement would be the use of a more customisable framework for machine learning. Scikit learn is very powerful, it is a black-box to users and thus actions such as modifications are difficult to implement. In addition, it is recommended for users with limited time or unrestricted resources to seek for other frameworks; this is due to the lack of GPU acceleration.

6 Conclusion

Optimising performance by incorporating context defining concepts and the ability to compute new classifiers without the necessity of training the model were the two main goals in this thesis. Through the builing of a separate CC, based on most frequently used words in user-submitted reviews, more data could be extracted from the original data set. The aim for higher performance of the baseline classifier was not achieved with the additional data from the CC. Due to the low performance of the CC, the extracted data was deemed less valuable than anticipated. Since the CC was trained on limited data, performance was reasonable. Even though the process of constructing the CC occupied a significant portion of time and effort dedicated to this thesis, the result is still a working CC which is essential to the workings of the SSE. A straightforward, yet well performing, approach for constructing new classifiers based of word similarity is the most valued part of the thesis. This approach offers the flexibility to construct new classifiers in real-time and eliminates the need for new data and model training. In spite of the increased flexibility, this approach comes with marginal sacrifice of performance.

With this as a foundation for further research, in the area of utilising CC in contemporary ways, new insights and questions arise for future researchers to answer. This could entail difficulties encountered here or original approaches in combining CC with alternative technologies.

References

[1] M.E. Stevens. *Automatic character recognition: a state-of-the-art report*. Technical note. U.S. Govt. Print. Off., 1961. URL https://books.google.nl/books?id=XF7faAzr0IwC.

[2] Russell A. Kirsch. SEAC and the start of image processing at the national bureau of standards. *IEEE Annals of the History of Computing*, 20(2):7–13, 1998. doi: 10.1109/85.667290. URL http://dx.doi.org/10.1109/85.667290.

[3] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[4] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[5] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[7] Jeremy Karnowski. Alexnet + svm, 2015. URL https://jeremykarnowski.files.wordpress.com/2015/07/alexnet2.png. Online; accessed INSERT DATE.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

[9] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.

[10] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. URL http://arxiv.org/abs/1312.6229.

[11] E. Santana, K. Dockendorf, and J. C. Principe. Learning joint features for color and depth images with convolutional neural networks for object classification. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1320–1323, April 2015. doi: 10.1109/ICASSP.2015.7178184.

[12] Julia Vogel and Bernt Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007. ISSN 1573-1405. doi: 10.1007/s11263-006-8614-1. URL http://dx.doi.org/10.1007/s11263-006-8614-1.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3063-3. doi: 10.1145/2647868.2654889. URL http://doi.acm.org/10.1145/2647868.2654889.

[16] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. In *NIPS*, 2012.