NTNU

Norwegian University of
Science and Technology

Tag Prediction in Social Media

Predicting Image Tags with Computer Vision
and Word Embedding

Petter Glad-Ørbak

Master of Science in Informatics
Submission date: May 2018
Supervisor: Herindrasana Ramampiaro, IDI

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Social Media produces vast amounts of user-generated content (UGC) every second, and images are increasingly part of enriching this content. The need for effective ways to organize and categorize content is bigger than ever. The proliferation of Big Data also offer new opportunities in regards to utilizing UGC in recommender systems. Considering the noisy and unstructured nature of user-generated text however, extracting valuable knowledge from it is not an easy task. Therefore, this thesis looks in the direction of images.

With the goal to extract some usable knowledge from these Social Media images, this thesis proposes a novel approach to predicting the tags and content of an image from Social Media with the help of *deep convolutional neural networks* (deep CNNs) and *word embedding* models.

A pre-trained model for computer vision is used to classify an image and extract predictions of its most likely content, and then evaluated against the image's tags to discover the model's tag prediction ability. Each of the predictions are used to produce similar syntactic and semantic information from a word embedding model. Using this aggregated information, the model's prediction ability is re-evaluated and performances are compared. In addition, the predictions are studied qualitatively to understand their degree of relevance.

The model is evaluated on a subset of the *MIRFLICKR25000* data set, which consists of 25000 images under the Creative Commons licence gathered from the Social Media platform Flickr. Although image auto-tagging is thoroughly researched, the task of tag prediction from images using computer vision and word embedding in this way is not done previously. The evaluation of this model on the data subset shows that comparable accuracy to state-of-the-art is achieved. Although they are not groundbreaking in terms of accuracy, results show a significant increase when expanding queries using a word embedding model.

Sammendrag

Sosiale medier produserer store mengder brukergenerert innhold hvert sekund, og bilder er i stadig økende grad del av denne. Med Big Data kommer nye muligheter for analyse og anvendelser av dette innholdet i anbefalingssystemer. Den økte datamengden medfører også økt behov for effektive metoder for kategorisering og lagring. På bakgrunn av brukergenerert innholds ustrukturerte og støyrike natur, er det heller ingen enkel oppgave å utvinne verdifull kunnskap fra den.

Med mål om å utvinne brukbar kunnskap fra disse brukergenererte bildene, foreslås det i denne oppgaven et system for prediksjon av emneknagger og innhold i bilder fra sosiale medier, ved hjelp av *dype nevrale nettverk* og modeller for semantisk vektorisering av ord (*word embedding models*).

En ferdig trent modell for datasyn brukes til å klassifisere et bilde og hente prediksjoner for mest sannsynlig innhold, hvorpå disse matches mot bildets emneknagger for å oppdage systemets evne til emneknaggprediksjon. Deretter brukes hver av prediksjonene til å uthente syntaktisk og semantisk liknende ord fra en ordvektormodell. Ved bruk av denne sammensatte informasjonen evalueres systemets evne til emneknaggprediksjon på nytt og ytelsen sammenliknes. I tillegg studeres prediksjonene kvalitativt for å oppdage deres relevanse.

Systemet evalueres på en delmengde av et datasett bestående av bilder og deres emneknagger oppsamlet fra den sosiale medieplattformen Flickr. Å predikere emneknagger på denne måte er ikke gjort tidligere, men resultatene kan indikere at den er verdig av videre forskning. Evalueringen viser sammenliknbare resultater for treffsikkerhet med nyere forskning. Selv om systemet ikke tilbyr grensesprengene treffsikkerhet, vises tydelig effektiviteten av å bruke modeller for semantisk ordvektorisering på denne måten.

Preface

This thesis is submitted to the Norwegian University of Science and Technology in Trondheim (NTNU) as part of a Master's degree in Informatics, fulfilling the requirements of subject IT3902. The work was conducted between 2016 and 2018 at the Department of Computer Science (IDI).

I would like to thank my supervisor, Associate Professor Heri Ramampiaro at the Department of Computer Science for his guidance throughout the process. I also express my deep gratitude to my parents, who have been an unwavering support during my time as a student.

I thank my friends Alexander Mykland and Henrik Schmidt for valuable feedback on this paper.

Lastly, I thank my girlfriend: my constant source of encouragement and motivation.

Table of Contents

List of Tables

List of Figures

Abbreviations

Symbol	=	definition
ILSVRC	=	ImageNet Large Scale Visual Recognition Challenge
NLP	=	Natural Language Processing
UGC	=	User-generated Content
ANN	=	Artificial Neural Network
CNN	=	Convolutional Neural Network
PLSA	=	Probabilistic Latent Semantic Analysis
LR	=	Logistic Regression
RLR	=	Robust Logistic Regression
API	=	Application Programming Interface
CCA	=	Canonical Correlation Analysis
AI	=	Artificial Intelligence
CPU	=	Central Processing Unit
GPU	=	Graphics Processing Unit
ReLU	=	Rectified Linear Unit
POS	=	Part Of Speech
CBOW	=	Continuous Bag-of-Words
DL4J	=	Deep Learning for Java
POM	=	Project Object Model

Chapter 1

Introduction

As an introduction to this thesis, this chapter aims to provide some background information to explain some of the motivation and challenges regarding tag prediction. Section 1.1 describes some of the circumstances behind the appearance of the problem, and why it should be solved. Section 1.2 follows with a description of the problem, and what has been done to solve it. Section 1.3 presents the main objectives this thesis seeks to accomplish. Some limitations apply to this work, and are explained in Section 1.4. Lastly, Section 1.5 provides an outline of the rest of the paper.

1.1 Motivation

Social Media has taken a firm role in our everyday life. Instead of talking to each other, even if we are in the same room, we might send a Snapchat photo or write a message on an online chat platform. Since we are now to some degree living our lives online, it has become an important arena for advertising, business, marketing and research. Analyzing users' online profiles and interactions can provide a great deal of insight to their interests and personalities. User-generated content (UGC) is one of the types of data that can be analyzed to obtain some of this knowledge. Status updates, or *posts*, on Social Media often come attached with a useful set of metadata. User profiling techniques use these kinds of insights to help make effective personalized recommendations, which have valuable applications. For instance, the selection of which products to recommend in ads greatly increases the likelyhood of a purchase. According to a study by Barilliance on its clients using their recommendation engine, up to 31%[1] of revenues from E-commerce sites came from the personalized recommendation of products, therein displaying the economic benefits of UGC utilization.

There are difficulties concerning these techniques, such as the lack of ratings or other information in new users resulting in *cold starts*. This is a problem, because even if a user has not seen a movie or clicked on a product, that does not imply a *dislike* toward

[1] https://www.barilliance.com/personalized-product-recommendations-stats/

it. Techniques use different approaches to estimate these missing values. For example, *collaborative filtering* (Goldberg et al., 1992) employs the ratings of users similar to one self, in the way that if person A and B both like an item X, and person A likes item Y, then person B is more likely to like Y as well. Another solution has been to focus on utilizing other metadata to strengthen the profiles, *tags*, or *hashtags*, being one of them. Tags are metadata in the form of keywords, which describe the content. Tag-based user profiling uses the tags associated with a user's profile and has proven quite successful. I.e., Firan et al. (2007) uses a tag-based approach for song recommendation which yields significantly improved results. Another application example is Ruocco and Ramampiaro (2015), who extract geo-spatial features from user tags, increasing performance of event-related image retrieval.

While user-generated text has a high degree of sparsity and unstructured nature, images are less unpredictable and can be expressed quite easily by three-dimensional vectors. Also, images from Social Media tend to have a reasonable degree of quality. One reason for that might be that our online user profiles seem to influence and reflect on our social status in real life. They are produced in enormous quantities as well; in just one day, over 95 million[2] photos and videos are posted on Instagram[3] alone. On Flickr[4], the number reaches about 25 million[5] photo posts on a busy day. Therefore a way should be found to extract knowledge from and organize these masses of visual content, so they can be more easily searched and retrieved. However, even though our brains make vision seem easy, it is a difficult and complex problem to solve for computers. Flickr for example has its own auto-tagging function which has run into some problems in the past, tagging images with controversial tags[6].

For the purpose of overcoming some of the challenges discussed above, this thesis' main contribution is an exploratory approach to predicting the tags of an image from its visual content. Image classification is first performed using a pre-trained deep neural network model, and then an attempt is made to increase prediction ability using a word embedding model to expand the predictions. The predictions are tested against the user-generated image tags from a subset of a research dataset for evaluation. The information offered by such a system is likely of interest to businesses, advertizers and researchers who use tag-based user profiling, and as another application, the tags suggested can be used in auto-tagging to categorize and organize photos without knowing their tags beforehand to enhance search and retrievability.

1.2 Context

1.2.1 Tag Prediction

The proliferation of Big Data has opened the eyes of researchers everywhere to the possibilities that lie in predicting the future using UGC and other forms of Social Media

[2]https://louisem.com/152018/instagram-stats-2017

[3]https://www.instagram.com/

[4]https://www.flickr.com/

[5]https://expandedramblings.com/index.php/flickr-stats/

[6]http://www.independent.co.uk/life-style/gadgets-and-tech/news/flickr-s-auto-tagging-feature-goes-awry-accidentally-tags-black-people-as-apes-10264144.html

data. As Schoen et al. (2013) mention in their paper, articles concerning predictions have gone from 0 to 18% between 2006 and 2012 at the WWW, ICWSM and IEEE SocialCom Conferences. While some motives are business oriented, there are practical real world applications as well, from predicting outbreak of diseases to elections and stock markets. It may even possible to predict nature catastrophies and provide earlier warnings. Tkachenko et al. (2017) for example, investigate the correlation between an increased use of certain image hashtags and real life flooding events. Other motivations include increasing search and retrievability, avoiding incorrect user tags and being able to discover events and trending topics.

Tag prediction is the task of predicting a set of tags given some content. It is closely related to that of *automatic image annotation*, especially in the research field, where the line between them seems to be largely obscured. One difference worth pointing out is that automatic image annotation's main goal of attaining categorization opts for tags at a higher semantic level, rather than accounting for the users' individual differences in tag conceptuality. This noisy nature of wild user tags makes tag prediction a truly challenging task, and remains an unsolved problem. Some of these challenges are discussed at length in Section 3.3.

Tag prediction is predominantly a natural language processing (NLP) problem, and a well researched area within the Twitter domain. For *tweets* (short textual Twitter posts), many approaches use topic modeling and latent Dirichlet allocation (LDA) (Godin et al., 2013), which are techniques for discovering the abstract topics appearing in a text. Convolutional neural networks (CNNs) which are discussed later in this thesis have been used as well (Weston et al., 2014). In collaborative tagging systems, probabilistic modeling (Yin et al., 2010) and nearest neighbor (Budura et al., 2009) are some pronounced examples of effective approaches. Zhang et al. (2012) also look at tag temporal usage patterns. As images have become common in UGC, research utilizing them for tag prediction has increased the past years. Early attemps using probabilistic latent semantic analysis (PLSA) (Monay and Gatica-Perez, 2003, 2004), then shifting toward similarity metrics and nearest neighbor-approaches (Zhang et al., 2006; Makadia et al., 2008; Guillaumin et al., 2009) before arriving at the paradigm of deep neural nets and image classification which today is the load-bearing column in regards to extracting information from visual content. Usually, works tend to pair a classifier together with other algorithms, i.e. robust logistic regression (RLR) (Izadinia et al., 2015), or provide a method of modeling the noisy user labels. Recent related works are thoroughly discussed in Section 2.1.

The work in this thesis attempts to predict one or more tags of an image gathered from Social Media. What constitutes the novelty of this approach is the combination of two techniques:

- Using computer vision to obtain base predictions
- Using a word embedding model to expand predictions

A vision-only approach is seemingly only attempted once (Park et al., 2016), and the combination with word embedding is equally rare. A few works use word embedding for representation (Murthy et al., 2015; Denton et al., 2015) and handling unseen labels (Li et al., 2015), but none as a form of query expansion as attempted in this work.

Definition of Tag

Most definitions of the word *tag* share a similarity; that it is a label designed to provide information about someone or something. In Social Networking and Media these are called hashtags because of the hash character **'#'** used in front of them. These tags allow for content categorization and marking which makes it available for later retrieval. However, since the problem of tag prediction is looked at from the angle of using Social Media, the fact that what is attemped to predict is something *user-generated* has to be considered. This creates a random variable in the process of predicting tags of content, because one has to deal with the individual's interpretation of what an object is or means. For example, a necklace to some might be only "necklace", while to others the label "heirloom" may apply as well. Hung et al. (2008) define tags as the *semantic concepts* that an object activates in a cognitive sense, which is the definition stuck to throughout the thesis.

1.2.2 Evaluation

To evaluate this system a dataset consisting of images and their belonging tags, preferably pre-processed, is needed. It is possible to crawl the data using Flickrs API, although it would take a considerable amount of time. Given the time and resource limitations, the MIRFLICKR25000 dataset[7] (Huiskes and Lew, 2008) is chosen instead. It is a research dataset consisting of 25000 images gathered from Flickrs public API, all licensed under the Creative Commons licence. The dataset also has complete manual annotations, raw tags, preprocessed tags (Flickr's automated pre-processor), pre-computed descriptors and software for bag-of-words based similarity and classification. The dataset is described in depth in Section 5.1.

1.3 Research Questions

The goal of being able to extract usable knowledge from user generated images is formalized into the following main research question:

RQ: *How to predict image tags from image visual content?*

To help answer this question and accomplish the research goals, three subquestions are defined:

RQ1: *How to predict tags using image classification?*

RQ2: *How to use word embedding models to help predict tags?*

RQ3: *What source of text is best for training the word embedding models?*

[7] http://press.liacs.nl/mirflickr/

1.4 Limitations

As this thesis and its work is conducted by a single person, there are some natural limitations in terms of time resources. Given the amount of hours needed to envelop one self into relevant research and theory behind it, it was decided to use a pre-trained model for image classification. In addition, the applications in the research field relevant to this thesis is generally highly demanding of hardware and computational power, which further cemented the decision. Since the work in this thesis is conducted in Java, there are also a few compatibility restrictions in regards to the chosen deep learning frameworks. For example, utilizing GPUs for computation and training is not yet supported for machines running Microsoft Windows as an operating system.

1.5 Thesis Outline

The thesis begins with **Chapter 1**, introducing the research field and the motivation behind predicting using UGC, a short presentation of tag prediction and this thesis' approach to solve the problem. The main objectives this thesis will answer are stated at the end. To give the work some perspective in the research field, a literature review on tag prediction is performed in **Chapter 2** and the research most closely related is discussed, summarizing at the end. In **Chapter 3** some theoretic background is provided on tag prediction and some of the challenges faced when using Social Media images in this approach to solve the problem are discussed. Then, some insights on the theory behind the techniques applied in this approach are given. **Chapter 4** describes the system and implementation in detail, before the experiements chosen for evaluation in **Chapter 5** are looked at. **Chapter 6** presents the results and discuss their significance, before summarizing with a conclusion and suggest possible directions of further work in the final chapter; **Chapter 7**.

Chapter 2

Literature Survey

To gain some perspective on the research field and its main challenges, a literature survey is performed on tag prediction and image annotation. First the works considered to be the most similar to the work in this thesis are discussed in section 2.1. Since tag prediction and automatic image annotation are named quite ambiguously and intermixed in the research field, it was decided to include both in this section for the sake of practicality. Other related works are detailed in in 2.2.

2.1 Related Works

The research considered most relevant to this thesis are works that perform image tag prediction or image annotation, where predictions are made using CNN-extracted visual features. Other works detail different methods of tag prediction or image annotation.

2.1.1 Image Tag Prediction

Tag Prediction at Flickr: A View from the Darkroom

Garrigues et al. (2016) present in their paper a large-scale system for image auto-annotation, with the goal of achieving better image search on Flickr. One of their main arguments is that the models trained from benchmark datasets like ImageNet (Deng et al., 2009) often come short when deployed to user platforms like Social Media. One of the reasons for this is that the majority of images in ImageNet are single-object images, and thus are unable to span the width of user concepts when they annotate their photos. Creating new datasets takes a huge amount of time and manpower, and is not a practical solution. Instead they argue the possibility of training models from scratch using user-generated tags only, resulting in an easier training process than the standard paradigm, ultimately without compromising performance. They choose the YFCC (Yahoo Flickr Creative Commons) (Thomee et al., 2015) dataset for training, as it is large in size and has tags that are solely user-generated. To meet the problem of the noise present in user tags, Garrigues et al. look

specifically at tags relevant to Flickr users. To account for the absence of click logs from personal image search, they visualize a substitution of (query, click) pairs with (photo, tag), finding the 10,000 most frequently used tags. Those considered irrelevant are removed (lanuage redundancies, locations, numbers and potentially offensive tags), resulting in a vocabulary of 4,562 tags on which to train their classifier. In design of their model YFNet, Garrigues et al. also deal with the problem of demanding computational costs and model sizes of very deep CNNs which make them impractical for smaller devices and embedded systems. Inspired by He and Sun (2014), they use a layer replacement strategy, factoring larger 3×3 filters to two asymmetrical 3×1 and 1×3 ones as shown by Szegedy et al. (2015). Reducing a 3×3 to an asymmetric double layer of 3×1 and 1×3 reduces complexity from $3 \times 3 = 9$ to $3 \times 1 + 3 \times 1 = 6$, resulting in a 33% complexity decrease without losing accuracy. They also use a *spatial pyramic pooling* (SPP) (He et al., 2014) layer in place of the last convolutional.

For evaluation, Garrigues et al. first extract subsets of the COCO (Lin et al., 2014) and VG (Krishna et al., 2016) datasets containing the overlapping concepts between their prediction vocabulary and the datasets (respectively denoted COCO67 and VG903) to serve as validation sets. Using training samples of 1000, 2000 and 4000 from YFCC, they test both pre-training on ImageNet before finetuning is done as in the standard paradigm, and training from scratch. Their results show that the accuracy of their model trained from scratch is similar to that of the standard paradigm, convergingly so for the highest amount of training samples. In addition to achieving the same mean average precision (mAP) on VG903, there was only a 0.43% difference on COCO67, thus demonstrating the validity of their argument that it should be possible to train effective classifiers from scratch without clean data.

Secondly they evaluate their models ability to act as a pre-trained model, fine tuning with both COCO and VG, before evaluating on COCO67 and VG903 respectively. The results show a significant increase in mAP when fine-tuning to the relevant domains, and they also demonstrate that pre-training on YFCC rather than ImageNet gives better performance when the number of training samples reach 4000. It is however worth noting that the number of ImageNet training samples remained constant at 1000. It is reasonable to think that an increase in ImageNet training samples would yield a similar improvement in accuracy. The authors also mention that they did not try training the Inception-v2 and Inception-ResNet-v2 architectures from scratch on the 4000 training sample YFCC dataset due to computational reasons. Also noted is that they mention recall as possibly of higher importance than precision regarding personal photo search, but never compute it or give a reason why they choose not to.

The authors conclude that their objective of training a state-of-the-art-performing classifier from noisy labels that also performs well in commercial deployment is met, but that there still is a significant gap in performance that needs closing. They also argue that training from noisy labels from datasets like YFCC and evaluating on VG is a suitable benchmark for further research.

Deep Classifiers from Image Tags in the Wild

Izadinia et al. (2015) present a method for tag prediction using a classifier trained on *wild* (user-generated) tags, and perform an analysis on the Yahoo Flickr Creative Commons

(Thomee et al., 2015) dataset. They like Garrigues et al. (2016) voice the important argument that current datasets lack the ability to grasp concepts and categories important to users (e.g. scenes, objects, attributes, activities, visual styles), thus proposing the use of wild tags in training classifiers. To better understand the user-generated wild tags, they perform an analysis on YFCC. Aiming to highlight some of the shortcomings of ground-truth-annotated datasets, the authors compare the YFCC dataset to ImageNet, and look at some statistical properties of the tags they contain. Looking at the 100 most frequent tags in YFCC, they estimate that ImageNet is missing roughly half of the most used Flickr tags. This is a surprisingly high number, and clearly demonstrates the conceptual gap between users' interests and current benchmark datasets. Other remarks by the authors confirm common suspicions, for example that user tags are highly ambiguous and map to multiple real-world concepts.

To meet the problem of noisy user-generated tags, Izadinia et al. introduce a stochastic EM (Expectation-Maximation) (Cappé and Moulines, 2009) approach to robust logistic regression for use in tag selection, performing prediction for each tag individually. In addition, they include a bias which allows calibration and easy adaptation to new datasets, or domains. Several models are then trained on the YFCC with different variations of the tag selection algorithm, some fine-tuned using a subset of NUS-WIDE (Chua et al., July 8-10, 2009) annotations.

For evaluating their models, Izadinia et al. put them through three separate tasks. First is tag prediction, where a baseline model using logistic regression (LR) is measured against one with robust logistic regression (RLR). A 200,000 image subset of YFCC is chosen for validation, predicting the 5 most likely tags for each image. For the 5, precision, recall and F-score are computed. The results show that while RLR precision is similar to LR, recall and F-scores respectively gain 17.1% and 8.4%. Next the authors evaluate the ability to objectively annotate images. Here NUS-WIDE is used for validation, as it is objectively and manually labeled according to 81 concepts. The models trained solely on user-generated tags give high gains on recall and F-score, with precision matching that of reported state-of-the-art (Gong et al., 2013). Training on clean NUS-WIDE data gives better results than training on YFCC alone. However, combining YFCC training and the calibration step ultimately produces the highest scores. The authors also show that their method of training on user-generated tags and calibrating on NUS can potentially reduce training costs by a factor of 200 while retaining performance, given the small amount of samples needed for calibration. Finally the models are tested on tag-based image retrieval. Good performance is given for single-tag queries, but here the NUS-WIDE trained model scores highest.

Izadinia et al. conclude that models trained on user-generated tags can be useful despite their noisy nature, especially paired with a small calibration, and that these wild tags are still a largely untapped resource.

HARRISON: A Benchmark on HAshtag Recommendation for Real-world Images in SOcial Networks

In their work, Park et al. (2016) present HARRISON (from the title), introducing a dataset consisting of Instagram images and their user-generated tags, with the goal of introducing a more suitable dataset for hashtag recommendation in the Social Network arena. The

dataset consists of 57,383 images, and around 165,000 unique hashtags. They observe that the top 1000 most frequently used tags form 59% of total hashtags, and therefore choose this subset as dataset classes, serving as ground truth. Park et al. also design a baseline model for hashtag prediction using a CNN classifier. The model is then evaluated on the dataset.

As Garrigues et al. (2016) and Izadinia et al. (2015) both discuss, there are a large amount of frequently used concepts in Social Media content missing in ImageNet. To help solve this problem in regards to hashtag recommendation and bridge the dataset gap, the authors employ two separate models, each performing a different category of feature extraction. They use the pre-trained image classification model VGG-16 (Simonyan and Zisserman, 2014) for both models, training for *object recognition* (Simonyan and Zisserman, 2014; Szegedy et al., 2014; He et al., 2015) on the ImageNet dataset, and *scene recognition* (Xiao et al., 2010; Patterson et al., 2013; Zhou et al., 2014) on the Place Database (Zhou et al., 2014). The scene recognition classifier allows extraction of important features missing in benchmark datasets like ImageNet. The two types of features are then fed to a separately trained multi-label classifier consisting of two fully connected layers and a sigmoid cross entropy layer, where hashtag probabilities are sorted and selected.

For evaluation, each model is first tested separately, then combined at the task of hashtag recommendation. Note that hashtag *recommendation* as described by the authors equates to tag prediction and Social Media-angled image annotation tasks in the research field. Hashtags are recommended for each image in the dataset, and precision@1, recall@5 and accuracy@5 measures are computed for the whole corpus. Precision@1 means the fraction of times their highest probable tag is among image user-tags. Recall@5 is the fraction of matches of top 5 predictions among image user-tags, and accuracy@5 is the fraction of times there is atleast 1 overlap between top 5 predictions and image user-tags. Results show that object recognition better spans user concepts than scene recognition when recommending or predicting user-generated tags. They also show that combining the two increases precision by 6.17%, recall by 2.57% and accuracy by 3.46% compared to object recognition alone. The authors report a precision@1 of 30.16%, recall@5 of 21.38% and accuracy@5 of 52.52%. Since they evaluate on a novel dataset which in turn is gathered from Instagram (Flickr is more commonly used), the distribution of user-tags may differ from other datasets, and affect the results to an unknown degree.

Park et al. conclude that their model performs well at short, descriptive tags, but is unable to suggest inferential tags and fine-grained classes. They call for further attempts using object detection and suggest combining with NLP techniques to increase contextual understanding as possible improvement options.

Automatic Image Annotation using Deep Learning Representations

Murthy et al. (2015) propose models for *image annotation* using CNN features and word embedding models, with the goal of achieving reliable automatic image annotation to better search and retrieval of images and videos. Murthy et al. also seek to demonstrate viability of CNN-extracted visual features over handcrafted, which are bottlenecks in regards to designing scalable real-time systems. They use three variations of *canonical correlation analysis* (CCA), and attempt a linear regression-based CNN model. They evaluate their models on three seperate datasets, and compare their performance to other approaches in

the research field. This work is interesting to this thesis in the way that they employ word embedding models as a tool in their approach, although used differently.

The authors use the pre-trained VGG-16 model to extract the 4096-dimensional visual feature vectors from each image. For representing each tag of the vocabulary, they utilize the word embedding model Word2vec (Mikolov et al., 2013a), resulting in a 300-dimensional vector for each tag. Then they apply three variations of CCA, which produces projections of the two vectors onto a plane with dimensionality less than or equal to the vector with the smallest dimensionality, while maximizing the correlation between them. Since standard CCA can only model linear relationships, kernel CCA is performed to include non-linear relationships. When a new image is tested, its visual feature vector is extracted and projected to the plane, and tags most closely associated with the closest matching visual feature are then selected. The third CCA variation is with *k-nearest neighbors* clustering, choosing k samples from each cluster, calculating their probability and ranking them accordingly. In addition, Murthy et al. train a regression-based CNN model where the last layer of the model is replaced by a projection layer, and the predictions are word embedding vectors.

For evaluation, Murthy et al. test their models on the Corel-5K (Duygulu et al., 2002), ESP Game (von Ahn and Dabbish, 2004) and IAPRTC-12 (Grubinger et al., 2006) datasets. For each dataset, precision@5, recall@5 and F-score are computed over the whole corpus, also including N+ (number of non-zero recall). Their results demonstrate that CCA is an effective approach to image annotation, achieving comparable measures on IAPRTC-12 and outperforming state-of-the-art on Corel-5K and ESP Game. The authors also show that using *word embedding vectors* over *binary vectors* gives a significant boost for all measures by including a version of their best performing model using binary vectors.

Murthy et al. conclude that CNN-extracted visual features perform comparably or better than handcrafted ones that limit scalability and use computationally expensive metric learning. In addition, they demonstrate the usefulness of utilizing word embedding models in image annotation tasks.

User Conditional Hashtag Prediction for Images

Denton et al. (2015) introduce a novel approach of predicting tags with the use of a CNN and user modeling. Unlike other works, their aim is to demonstrate that utilizing user metadata is an effective method of improving performance of tag prediction. Using CNN visual features and user metadata, two techniques for combining them to a learning are explored. In evaluation of their framework the authors perform tag prediction on a dataset consisting of de-identified posts from Facebook.

Users tag content differently, and mappings from real-world concepts to tags are endless. Denton et al. mention continuous change in user interests, feelings and bias toward specific tags as examples of this type of noise in tags. The authors' approach to this problem is exploiting the large amount of available user metadata to model each user. For representation Denton et al. propose a model for embedding images and tags to a joint space as in Weston et al. (2011). For images, visual features (*image descriptors*) are first extracted from a CNN trained on approximately 1 million Facebook-images and 1000 categories. Then the descriptor is used in the embedding process, where the authors describe

three methods. A simple linear mapping of an image descriptor without any user information (*bilinear*), an additive which adds a user dependent bias on top of the image descriptor (*user-biased bilinear*) and a multiplicative where the image descriptor is gated through the user descriptor (*user-multiplicative tensor*), so that each user feature vector produces a unique image embedding.

Denton et al. train and test their models on a large Facebook-gathered dataset of 20 million public images by an approximate 10.4 million de-identified users. They limit tag vocabulary to the 10K most frequently used in one version. However, they point out that the natural distribution of tag use inhibit prediction ability of less frequent tags, and thus create a second *balanced* version of the dataset. In this version the frequency of the top 500 tags are downsampled to that of the 501st. In the dataset, users are described by 4 metadata: age, gender, country and city, which are broken down to form 10-dimensional user feature vectors. The authors select a subset of 100,000 images for testing. Performance measures precision, recall and accuracy are computed for each image (@1, @10, @10 respectively).

Although performance measure results are fairly low, they show that the models applying user metadata significantly outperforms the frequency baseline and bilinear models. The two user metadata models achieve comparable results on the naturally distributed dataset, but the multiplicative performs better on the balanced. A too high value of dimensions for embedding negatively impacts performance. Denton et al. also demonstrate the qualitative performance of their predictions, showing that predicted tags are often relevant to the content if not equal to ground truth. The models trained on the balanced dataset also prove much more capable of predicting less frequent tags. To give better understanding of their results, the authors explore and provides some insight to which types of tags are most dependent on user information to be accurately predicted.

Denton et al. conclude that user metadata can be effectively used to better image tag predictions, and show that downsampling the most frequent tags in a dataset and creating a more balanced distribution may produce more varied predictions.

2.1.2 Other Works

Zero-shot Image Tagging by Hierarchical Semantic Embedding

Li et al. (2015) present their system HierSE for zero-shot image tagging. Zero-shot learning is a form of supervised learning where the goal is to classify labels not previously seen. Zero-shot image tagging therefore focus on being able to tag images with new concepts even though it was not included in training. Even though this work does not detail tag prediction per se, it arguably has the most resemblance to the work of this thesis. The main difference is that zero-shot image tagging try to label a single objective ground truth with as high accuracy as possible, while tag prediction is a multi-label problem. The approach is different as well, as the authors embed both images and labels to a semantic space. They project images onto the space by using a pre-trained image classification model, obtaining prediction labels of highest probability. A semantic embedding model subsequently assigns vectors in the space for the labels. If multiple labels are chosen, the vector for an image is a convex combination of each of the label vectors. The classifier then selects the vector with the highest *cosine similarity* to the image vector. Then Li et al. improve their system by eliminating the problem of missing concepts in word embedding models and

ambiguous labels (which normally produce the same vector regardless of meaning). As the image classifier is trained on labels from WordNet, a lexical database for the english language, they assume each predicted label has a node in the WordNet hierarchy. For this reason they should be able to trace the ancestors of the label. Vectors are then generated as a combination of the node and its parents, with close relatives having more influence.

Li et al. evaluate their models on the ImageNet (Deng et al., 2009) dataset, and include 1,548 unseen labels which are parent nodes and children of the 1,000 training labels. As some labels of the WordNet hierarchy are phrases, the authors decide to allow *partial matches*. This means splitting the labels that are phrases to single words before embedding them, and look for single-word-matches. Two word embedding models are trained using different sources of text. The latest *Wikipedia* dump of 2.2 million words for one, and user-generated tags from 4 million Flickr images for the other. In addition, they experiment with the pre-trained Google News model (Mikolov et al., 2013a). They choose accuracy percentage at 1, 2, 5 and 10 guesses as a performance measure. All their hierarchical models outperform previous works by a fair percentage, their best performing one using the Flickr-trained embedding model achieve close to double accuracy at 1 guess.

The authors conclude the effectiveness of including the WordNet hierarchy to their semantic embedding framework, and their two good practices of introducing partial matching and training of word embedding models on the user-generated tags from Flickr images.

2.2 Summary

Through this chapter, a few important works on tag prediction and image annotation has been discussed. There are other works which could be included as well, but given the constant developments of new technology used by current state-of-the-art approaches it was decided to focus on works from recent years. As the review shows there are many different approaches to the two tasks. Most of the successful approaches combine the use of convolutional neural networks (often using pre-trained models) with other disciplines. Common for all however is that performance measures indicate there are still much room for improvements, both in terms of developing more lightweight architectures and increasing prediction ability. Another insight is that there seems to be some difficulty in comparing the performance measures of works. Reasons for this may be:

- Classifiers use different architectures to fit the need and scope of the application.
- There is a high variety of datasets being used, both for training and testing, which may have oscillating effects on results.
- Performance measures are computed differently.

This may all indicate the need for a new and clearly defined baseline for tag prediction approaches. The effects of dataset variety however will be hard to remove from the equation, as they greatly enhance performance toward specific domains and facilitate real-world deployment.

Among the reviewed works are some insights and discoveries especially interesting to the work of this thesis. Since there are large semantic gaps between benchmark datasets

like ImageNet (Deng et al., 2009) and user concepts in Social Media, Izadinia et al. (2015) and Garrigues et al. (2016) demonstrate the performance increase of training the image classifier on user-generated tags for tag prediction purposes. Li et al. (2015) showed that vectors produced from training a word embedding model on user-generated tags from Flickr images are semantically closer to tag prediction labels than other, larger models with text from the Web. This particular insight is highly relevant to training well performing word embedding models for tag prediction. The technique of partial matching also highlighted by Li et al. will be useful when dealing with phrases. It is noted that statistical modeling shows good results as well, but is beyond the scope of this thesis. Table 2.1 shows a brief summary of the main ideas from each reviewed work.

Author(s)	Main ideas
Garrigues et al. (2016)	*Design a lightweight CNN architecture which they train on user-generated tags for the purpose of making reliable lightweight, consumer-applicable image annotators for personal photo search.*
Izadinia et al. (2015)	*Perform an analysis on the Yahoo Flickr Creative Commons dataset to understand the span of user concepts and usage of wild (user-generated) tags. They train an image classifier on wild tags which they use together with robust logistic regression for tag prediction and image annotation.*
Park et al. (2016)	*Introduce a new benchmark dataset, HARRISON, for hashtag recommendation. They then train an image classifier on datasets for object and scene recognition, and perform hashtag recommendation on their dataset.*
Murthy et al. (2015)	*Perform image annotation using a pre-trained image classifier, and uses statistical modeling (canonical correlation analysis) to maximize correlation of image features and tags as they are embedded to a joint space.*
Denton et al. (2015)	*Propose a method of tag prediction utilizing the large amounts of available user metadata. They use an image classifier to obtain an image descriptor, and perform embedding of image descriptors and tags to a joint space. User data is used to produce a user descriptor which is paired with the image descriptor in the embedding process.*
Li et al. (2015)	*Perform zero-shot image annotation and introduce the hierarchical structure of WordNet in semantic embedding of images and tags to tackle novel labels. Experiment with word embedding models trained on different data sources.*

Table 2.1: Table summarizing the main ideas of reviewed works

Chapter 3

Theory

In this Chapter some theory is provided on tag prediction and the technologies used in this approach. This will give a better understanding of the most central topics discussed throughout the thesis. 3.1 begins with an introduction on *computer vision*, and its best performing technologies to date. In 3.2 is a brief discussion on what *natural language processing* (NLP) is and the subdiscipline of *word embedding*, which is used in this approach. The chapter ends with 3.3, where some of the challenges of tag prediction are discussed, including those induced by the technologies.

3.1 Computer Vision

As humans, most of us are gifted with a powerful tool. Together our brains and eyes are able to distinguish and classify real world objects in tens of thousands of categories without thinking twice. Vision in itself and using our brain to interpret what our eyes take in may be something of a triviality to us, but is an important and complex problem to solve for computers. Computer vision is the research field of making a computer see and interpret like humans, in order to gain high-level information from images and video. The goal is ultimately for systems to be able to perform the same visual tasks humans do in an automated fashion, which proves a difficult task for a number of reasons. First of all, vision seems to be about knowledge as much as the eye itself, and the understanding is made by using both deductive and inductive reasoning. Even though computers can easily be equipped with enough sensors to perceptively surpass humans, inductive reasoning and also understanding the context of what is seen is proving difficult to replicate for computers. Another obstacle is simply the vast number of categories and concepts related to the real world. Although the study of computer vision emerged in the 1950s, it would take quite some time before it made serious progress. Like in many other areas of early computer science, the task was greatly underestimated. For example, in 1966 Marvin Minsky (co-founder of MITs AI Lab) assigned a pair of students over the summer to link a camera to the computer and have it describe what it sees. The time table proved to be a a litte too small.

Great progress has however been made the past decade in regards to computer vision performance, much attributed to advances in hardware and machine learning. The introduction of the *graphics processing unit* (GPU) in particular unlocked the potential of applying *neural networks* to problem solving. A type of neural network named *convolutional neural networks* (CNNs) (Lecun et al., 1998), which we discuss in Section 3.1.2, have proved excellent for computer vision applications. CNNs have been applied to these tasks for a while now, performing a variety of them such as recognizing house digits (Sermanet et al., 2012) and traffic signs (Sermanet and LeCun, 2011). Even though earlier performances were decent, the introduction of the much larger and better annotated *ImageNet* (Deng et al., 2009) dataset made for great advances on state-of-the-art accuracy on *image classification* and *object recognition* tasks (Krizhevsky et al., 2012). Google's own CNN model Inception-v3 (Szegedy et al., 2015) reports a 3.5% top-5 error and 17.3% top-1 error on the validation dataset of the 2012 ILSVRC (Russakovsky et al., 2015) competition, where the task is to categorize a subset of the ImageNet dataset into 1000 categories. This pre-trained model will be used to perform image classification, retrieve information from the visual content and form the basis for the predicted tags. Since this system directly uses the predictions produced by the classification, it means the accuracy of the system will depend largely on whether it could detect an object or more from these 1000 categories in the photo. There are several other models which we could also use (e.g. AlexNet, VGG-16), but as of now Inception-v3 is the most balanced in terms of accuracy versus computational cost, and is made easily available.

Though the performance of computer vision on classification and object recognition tasks are generally good, training of the models is a computationally complex process and takes a significant amount of time. Therefore, given the thesis' limitations in terms of time and resources (mainly hardware) it is decided to use the pre-trained Inception-v3 model rather than training one from scratch. In the further work Section 7.1.1 some improvement suggestions for the model are made. In addition, the model training requires a substantial number of training samples to avoid overfitting. The ability to train accurate models relies on the availability of large annotated datasets, and the existing benchmark ones arguably make for insufficient models when applied to specific domains, i.e. Social Media platforms as Izadinia et al. (2015) discuss. ImageNet for example is often restricted to single-object images, which means they cannot properly span the width of user concepts in regards to predicting image tags. These datasets are gathered from various Social Media platforms or the Web, then manually annotated. As this is a highly time demanding and labor-intensive process, expanding a dataset is difficult.

3.1.1 Artificial Neural Networks

As an introduction to artificial neural networks, a brief timeline of milestones in the field is provided. In 1943 McCulloch and Pitts (McCulloch and Pitts, 1943) introduced a computational model for neural networking, marking the beginning of the research on the subject in regards to artificial intelligence. Other important milestones in the field of neural networking include:

- Rosenblatt's *perceptron* (Rosenblatt, 1958) - a pattern recognition algorithm
- Paul Werbos *backpropagation algorithm* (Werbos, 1974)

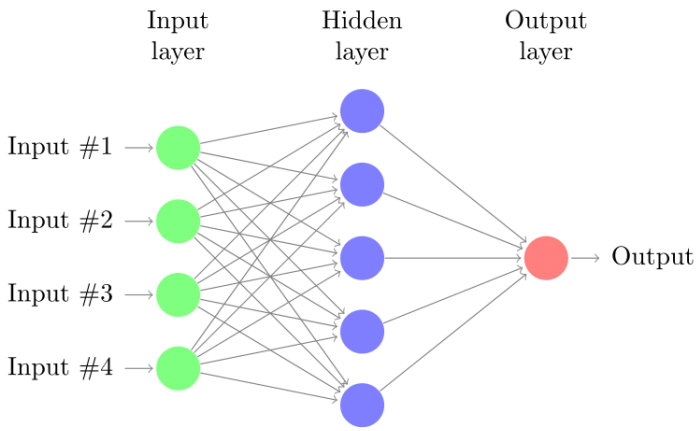

Figure 3.1: Example of an artificial neural network

- The introduction of *convolutional neural networks* (Lecun et al., 1998)
- *Deep belief networks* (Hinton et al., 2006)

Artificial Neural Networks, or ANNs, are biologically inspired computation systems consisting of layers of nodes, where the thought originally was to mimic the way our brains learn by forming strong connections. The first and last layer are input and output layers respectively, while the ones in between are called hidden layers. In Figure 3.1[1] a 3-layer network is depicted. The nodes connect to each other across layers, and each has an activation function similar to biological neurons. This function takes the sum of weighted input signals from the previous layer and computes what output signal is to be forwarded to nodes in the next layer. Figure 3.2[2] displays what is considered the modern anatomy of an artificial neuron. The way the networks learn is by being fed huge numbers of examples and correct answers, each time backpropagating errors through the layers of nodes. The backpropagation process adjusts weights at every node to gradually produce a more correct output.

These networks can be applied to almost all kinds of problems, but as mentioned they require a huge number of training samples in order to learn. *Overfitting* is a trait that can arise if the training samples are too few or in some way flawed. Overfitting is when the training data results in poor generalization knowledge for the network, ultimately yielding poor performance on untrained data. Common applications today are image classification, driverless vehicles, robots, face and speech recognition, artificial intelligence for games and more.

[1] http://www.texample.net/tikz/examples/neural-network/
[2] https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c

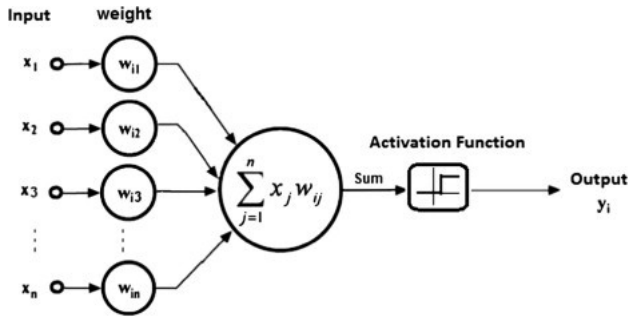

Figure 3.2: The anatomy of an artificial neuron

3.1.2 Convolutional Neural Networks

In their paper, Lecun et al. (1998) introduce convolutional neural networks (CNNs), which are large, complex variations of ANNs that work especially well on computer vision tasks because of their structure. The networks use multiple replications of the same neuron, and are often deep, meaning they have multiple hidden layers. For computer vision tasks, images are represented as a matrix of pixel values. In the *convolution* layer, the CNNs use smaller 3×3 matrices called *filters* to slide (also denoted a stride) over a small part of the image matrix. Element-wise multiplication is performed, resulting in a 3×3 *feature map* matrix of higher level features. This computation process is called a *convolution*, and is how the network performs feature extraction. This computation happens in the convolution layer, together with an operation where the negative values of the feature map are replaced by zeros to add *non-linearity* to the network. The operation is called *rectified linear unit* (ReLU), and is implemented due to the fact that most real-world data is non-linear. Another type of layer is *max pooling*, where dimensionality of feature maps are reduced while retaining the most important information. A sliding operation similar to the convolution is performed, but instead of a multiplication, the maximum values of each stride are retained. Finally, data is fed to a *fully connected* layer, before the output labels are inferred from extracted image features using a classification algorithm. Figure 3.3[3] demonstrates how a deep 2-dimensional CNN may look like. $X_{n,m}$ represent inputs, which are fed into a convolutional layer of A-neurons. Next is a max pooling layer, before another convolutional layer of B-neurons. F is the fully connected layer where outputs are computed.

Performance on object detection tasks and image recognition really started to soar after the introduction of the ImageNet (Deng et al., 2009) dataset. A well annotated dataset finally provided the networks with sufficient examples. Groundbreaking accuracy was first achieved by Krizhevsky et al. (2012) with their CNN architecture *AlexNet*, leading to a period of continous improvement on the ILSVRC challenges, eventually surpassing even human accuracy (around 5% top-5 error).

Since CNNs hold the best results on computer vision challenges, improvements in the field are primarily related to making these networks better. One of the drawbacks of these

[3] http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

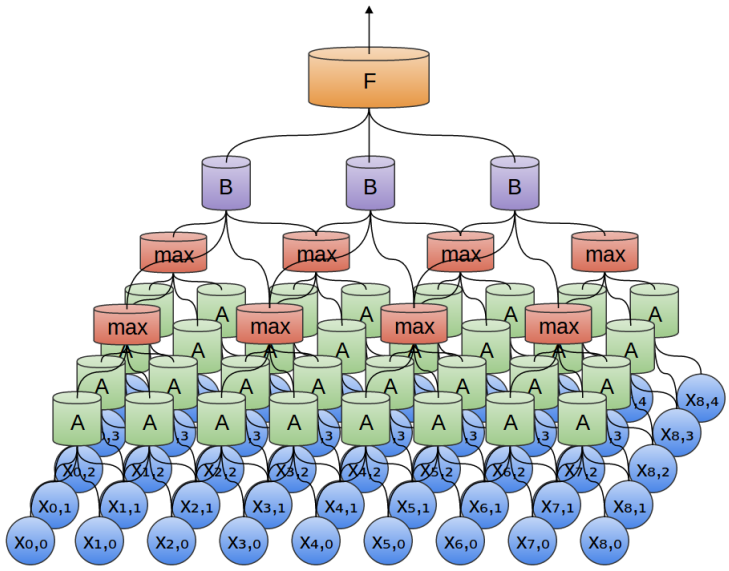

Figure 3.3: A deep two-dimensional CNN with max pooling layers. $X_{n,m}$ represent inputs. A and B are convolutional layers, and max are max pooling layers.

deep networks is that they become quite large in size and demanding of computational power, which limit their portability to smaller devices and embedded systems. Given these premises, improvement focus has been on making the networks smaller and reducing computational complexity, ideally without any significant sacrifice in accuracy. Han et al. (2015) for example show that it is possible to considerably reduce size and complexity of networks using their 3-step pruning method, while fully retaining accuracy. They first study the network to discover what connections are important, before pruning them and retraining the weights of the remaining network. Their method reduces parameters in AlexNet and VGG-16 by a respective 9 and 13 times. Another important example is Rastegari et al. (2016) who show that using binary filters and binary input to convolutional layers can make convolution operations in networks up to 58 times faster, and use 32 times less memory, also without losing accuracy.

3.2 Natural Language Processing

Natural language processing (NLP) is an interdisciplinary field within artificial intelligence where the goal is to have computers understand, analyze, process and manipulate human language. Computers already understand a language of their own, which has its own set of syntactical rules. To be understood the computer instructions have to be correct to the letter. Their meanings are unambiguous and straight-forward, unlike human language which can carry rich underlying semantic descriptions in a single word. While desired action can be provided for all combinations of a computer instruction set, its vocabulary is

miniscule compared to ours. Sentences can be combined in an almost infinite number of ways, and carry hidden semantic meanings like sentiment, humor and sarcasm. Therefore we attempt to make computer models which can make some sense of what we are saying and how we are saying it.

Analyzing sentences by decomposing them to determine structure and each word's category in *part of speech* (POS) were considered important areas of focus in earlier research, but efficient part-of-speech tagging was difficult to attain due to the ambiguity of words in our language. Research up until the 1980s mostly consisted of generating complex hand-written rule-sets, before machine learning algorithms started to get traction. The first algorithms employed techniques like *decision trees* ultimately resembling the practice of using hand-written features, but the introduction of statistical modeling to the field marked a revolution and much better performance. Machine learning approaches using probabilistic modeling greatly helped alleviate the problem of word ambiguousness, and increased robustness, especially when encountering new or misspelled words. Hidden Markov models are examples of successful statistical introductions and are still used to date.

Language models are probability distributions over sentences, assigning a probability to each word in a sentence which allows comparison of relative likelihood of sentences. It also gives words a sense of context in the text. The *unigram model* is the simplest way of generating these, assigning each word with its own probability of occurence in the text. *N-grams* estimates the probability of a word based on the probability of preceding words; its *context*. *Skip-gram* is used by the word embedding models employed in this thesis, and tries to predict its context (preceding and succeeding words) from the word itself. The models have proven useful for *query likelihood* models and machine learning approaches to NLP problems, but are affected by the *curse of dimensionality*. In this case meaning that the number of possible sentences from combinations of words grows exponentially when vocabulary increases.

The success of artificial neural networks the past years on detecting patterns in large amounts of information and the proliferation of big data has sparked interest in deep learning approaches to NLP problems. Among these are *neural language models*, who avoid the curse of dimensionality by using a distributed representation of words as vectors in a continuous space and training an artificial neural network to predict the context of a target word. The process of vectorizing words is called *word embedding* and is discussed in the next section.

3.2.1 Word Embedding

Word embedding is a set of NLP techniques for mapping words into vectors consisting of real numbers. These are called *word vectors*, the mapping is performed in such a way that similar words have similar representations, so that semantically similar words end up in close proximity of each other in the semantic vector space. Techniques to perform word embedding are roughly divided into two groups; *count-based* methods and *predictive* methods. A simplified explanation is that the count-based method use word count statistics to look at co-occurences between words, and use them in the vector generation process. The predictive methods frame the process as a supervised task where the goal is to train an ANN to adjust the weights such that it maximizes probability of outputting

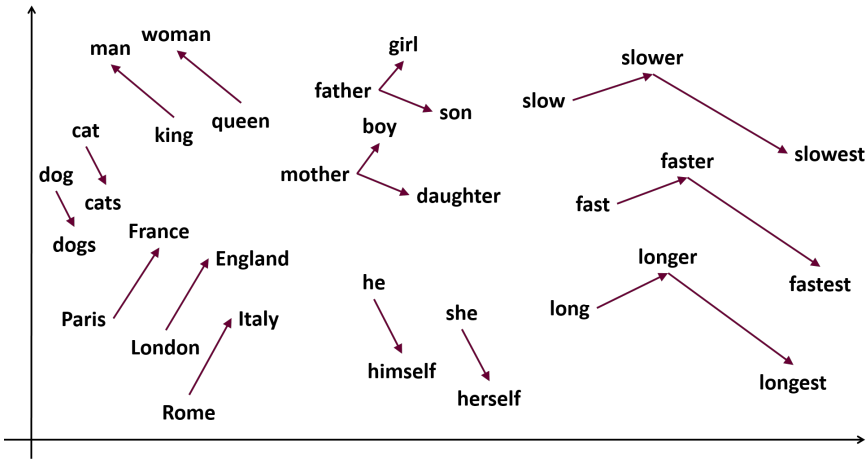

Figure 3.4: Words as points in a semantic vector space. Lines between them display relationships.

the correct context of the word as it was observed in the text. These predictive methods are also called *neural probabilistic language models* and were introduced by Bengio et al. (2003). The count-based embedding first generate vectors of a high dimension, then use dimensionality reduction techniques afterwards. The predictive methods represent words as vectors of word *features* which are actually the weights of the hidden layer in the ANN. Using this continuous distributed representation generates vectors of far fewer dimensions than preceding techniques, often in the tens or hundreds. Baroni et al. (2014) demonstrate the computational superiority of these approaches in their paper, justifying the hype of these models the past few years.

Word2vec

In 2013, then a researcher at Google, Tomas Mikolov together with his team introduced a set of models for word embedding known as *Word2vec* (Mikolov et al., 2013a). While several techniques of predictive word embeddings had been around for a while, these models were revolutionary in terms of computational efficiency, and opened opportunities for commercial use. Word2vec made it possible to train embedding models in a swift manner from huge corpora of unlabeled text with a great number of dimensions. In addition to their group of models, Mikolov et al. conducts an analysis on the properties of generated vectors, discovering that they were able to capture both syntactic and semantic relationships between words to such a degree that vectors can be used for logic reasoning. In their paper, examples of these relationships are illustrated by algebraic operations. For example, $vector(Uncle) - vector(Man) + vector(Woman)$ will produce a vector very close to $vector(Aunt)$. Figure 3.4[4] shows a number of words as points in a space, and the lines between them display the relationships between them.

Word2vec's architectures are three-layer shallow neural networks consisting of an in-

[4] http://www.samyzaf.com/ML/nlp/nlp.html

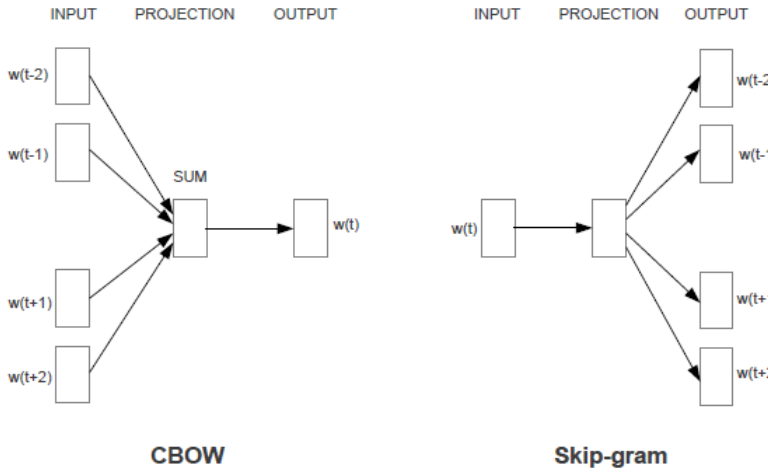

Figure 3.5: Continuous Bag-of-Words and Skip-gram where $w(t)$ denotes the target word, and $w(t \pm n)$ denotes the n words to each side of the target word.

put layer, a hidden layer (also called a *projection layer*) and an output layer. When iterating over sequences of text, Word2vec make use of a *context window* of a chosen size. The context window signifies the number of words to each side of the *target word*. Then, there are two different architectures which decide how Word2vec uses the words from the context window in the process of learning representations. The two are named *continuous bag-of-words* (CBOW), and *continuous skip-gram* which in all essence is the inverted version of CBOW. The CBOW approach utilizes all words from the context window, calculates their average and learns to predict the target word from them. Skip-gram on the other hand learns to predict the context words when given the target words as input. Figure 3.5 depicts an intuitive illustration of these architectures. While CBOW is a fair bit quicker to train and has higher accuracy for frequent words, it is not so good at predicting infrequent words. What is different about skip-gram is that it treats each target-context pair as a new observation, which ultimately works better for infrequent words. Another advantage that comes with treating each target-context pair as a new observation is that it works well even with relatively small amounts of training data, since it generates more training instances. Word2vec also has two separate training algorithms. The *hierarchical softmax* algorithm works best for infrequent words, while *negative sampling* is best suited for lower dimensional vectors and works best for frequent words.

These Word2vec models display excellent results, and the authors express excitement towards using them to help solve difficult NLP tasks like machine translation. They also mention the possibility of providing automatic extension for *knowledge bases*. For this thesis, we will utilize prediction tokens from the image classifier and retrieve word vectors of closest cosine similarity. The idea is to exploit the semantic and syntactic relationships that the models are able to capture, hopefully to increase the prediction accuracy. A weakness to Word2vec worth noting that is relevant to this work, is that any word not in the

vocabulary of a model will return an empty set of nearest words. Other word embedding models exist which also show great results on these tasks (e.g. *GloVe*[5]), but given the ramifications of this thesis, software compatibility and availability of pre-trained models, the decision fell on Word2vec.

3.3 Tag Prediction - Challenges

Images have become an increasing part of UGC the past years, and as a result, interest in tag prediction and automatic image annotation is greater than ever. The large online photo collections can only reach their full potential if they can be searched and retrieved, which calls for effective categorization methods. Given the increase in performance of recommendation systems and the inviting marketing prospects that user modeling offers, predicting tags from user content is also in high demand. However, there are multiple challenges to consider when trying to achieve accurate tag prediction. As the literature survey of Chapter 2 shows, the two main difficulties on the subject are *tag noise* and the semantic gap between objective and subjective.

3.3.1 Noise

In most Social Media environments the users themselves are responsible for tagging content. The act of tagging one's content should be self-encouraging in itself, as it allows for easier retrieval at a later point in time, but this is not always the case. A big problem of tasking users with the responsibility of tagging content is that they often simply refrain from tagging their content at all. Although user-generated tags may not always be accurate or make sense, the presence of tags are always favored to an absence. Garrigues et al. (2016) comment on this in their paper, with their own analyses showing that missing tags are the most common type of noise. So what motivates users to tag content? The tagging behavior of users is linked to several things. According to Strohmaier et al. (2010), tagging motivation can at least be divided into *categorization* and *description*. Users motivated by categorization practice high-level feature tagging for making retrieval at a later point in time as easy as possible, while those motivated by description try to most accurately describe the content. Those motivated by the latter often use a higher number of tags as well, which makes sense since the level of description should increase with the amount. Another study on content tagging by Nov et al. (2008) looks at what motivates photo tagging activity on Flickr. They divide motivations into *organization* and *communication* in a similar fashion to Strohmaier et al. (2010), with "communication" corresponding to "description" in Strohmaier et al. (2010). They put these motivations in context of audience (self, family & friends, public), finding that having a public audience correlates positively to level of tagging, and that "communication" is the strongest motivator of the two categories. The "self" also displays a correlation, although not as strong as the "public". "Family & friends" did not show any significant correlation, which might be explained from the fact that "communication" is the strongest motivator. It is reasonable to think that the describing of content to family and friends is carried out verbally or through chat rather than image tags.

[5] https://nlp.stanford.edu/projects/glove/

Figure 3.6: im46 from the MIRFLICKR dataset and its belonging tags.

In addition to missing tags, incorrect tags are a commonly encountered noise type. As users are tasked with the tagging, cases where personal associations to image content ultimately results in tags that have neither categoric nor descriptive sense are inevitable. A person may be biased toward a certain group of tags. Some users belong to specific communities which practice systematic tagging of photos using a sort of *tagging profile* which reflects on their environment, almost similar to branding. The fitness community on Instagram is an example of such a community, with images often dominated by tags such as "wod" (workout of the day), "fitness", "beastmode", "nopainnogain" (no pain no gain) which are not necessarily categorically or descriptively related to image content. Influencers and inspirational figures may also cause this type of tagging behavior.

It is apparent that the span of user concepts is enormous, and that their tagging vocabulary is consequently of equal proportions. Not only might personal associations dictate tagging behavior, tags also map to several different real-world concepts, making them highly ambiguous. A simple example is "break", which has over 70 different meanings[6]. Figure 3.6 illustrates the noise in user tags. Even though the photo only contains a pair of feet and slippers, it includes tags like "toronto", "50mm" and "explore". The difficulty in predicting these types of tags is not hard to understand.

So how does one deal with this noise? Some research works attempt to model the noise. Xiao et al. (2015) for example introduce a probabilistic model which explicitly infers non-noisy tags from noisy when training the CNN. An option is to disregard the noisy tags, but since they are of such numbers, the training data would be largely incomplete. Another interesting and rather successful approach has been to train classifiers on these user-generated tags. Garrigues et al. (2016), Izadinia et al. (2015) and Park et al. (2016) which were included in the literature survey are pronounced examples. An additional obstacle to consider with this approach is deciding on a vocabulary. The amount of unique tags in datasets with user-generated tags could potentially give vocabularies in the tens of thousands, and training classifiers of that dimension is not easily feasible. Finding what subset of these tags are most relevant is clearly the solution, but a difficult one to achieve.

[6]https://muse.dillfrog.com/meaning/word/break

3.3.2 Dataset Gap

Since image classification, tag prediction and image annotation methods rely on ImageNet training, Izadinia et al. (2015) conducts an analysis on the Yahoo Flickr Creative Commons (YFCC) and ImageNet datasets, and do some comparisons. What quickly became clear is that there is a large semantic gap between user concepts and those included in ImageNet. YFCC is gathered from the photo sharing site Flickr, and the analysis revealed that almost half of the most commonly used Flickr tags are missing from ImageNet. Even some of those included are poorly represented and would make poor training data. This clearly demonstrates that even if ImageNet gives excellent results for objective image classification and object recognition tasks, it makes for poor deployments to user platforms like Social Media. Missing concepts are from categories like activities, places, emotions, art and culture, scene types and so forth. Some of these missing categories are included in style (Murray et al., 2012; Karayev et al., 2013) and scene recognition (Xiao et al., 2010; Zhou et al., 2014) works.

A possible remedy to this conceptual gap is to augment or generate new datasets which make better models for consumer deployment. This is as earlier discussed a costly operation in terms of time and labor, and other solutions have yet to appear. This thesis' approach to the problem is to use a word embedding model to attempt to bridge the conceptual gap to some degree.

Chapter 4

Approach

This chapter walks the reader through the approach of this thesis, detailing each part of the process of performing tag prediction. The frameworks used for the implementation are briefly discussed in the first section. Section 4.2 entails the image classification part where visual features are extracted from image content, and some technical details about the pre-trained model. Section 4.3 explains some technical details about the word embedding models and how they are utilized. The final section demonstrates how the tag prediction itself is performed.

4.1 Frameworks

4.1.1 TensorFlow

TensorFlow[1] (Abadi et al., 2015) is an open source software library for numerical computations founded by Google. It can run on multiple platforms and both central and graphics processing units (CPUs and GPUs), either distributed or non-distributed. The manner of which TensorFlow executes computations helps in making machine learning and computation in deep neural networks more comprehensible. Code is written to create a *computational graph*, a data structure for describing computations and then executing it. These graphs make saving, loading and executing models an easy task.

For this thesis TensorFlow is used to load the Inception-v3 model as a graph, and execute *Tensors* on it. The Tensors are a data type and TensorFlow's own generalization of *n-dimensional arrays*. Images are read as *bytes* and created into Tensors. A small drawback of using TensorFlow is that utilizing the graphics processing unit (GPU) is not currently supported for Java when running on a Windows 10 operating system. Inferral on the Inception-v3 graph can be performed about three times as fast with a GPU instead of a central processing unit (CPU).

[1] https://www.tensorflow.org/

4.1.2 Deep Learning for Java

Deep Learning for Java[2] (DL4J) is a Java toolkit for deep learning, which supports a large number of algorithms. It relies on the computing library ND4J which is an equivalent to the *numpy* library for Python. It can be used for developing and training neural networks, and perform various computations and clustering. For this thesis DL4Js Word2vec models are used to train, load and interact with word embedding models.

4.1.3 Apache Maven

Apache Maven[3] is a project management tool which can handle project builds, and the import of libraries and their models simply by adding a few lines of text. Maven uses an XML file called *project object model* (POM) for managing dependencies and project configurations. New libraries are downloaded from the Maven Central Repository after they are added to the as dependencies in the POM.

4.2 Image Classification

One of the objectives of this thesis is to predict tags from image visual content. Image classification using convolutional neural networks (CNNs) is currently the state-of-the-art technology for extracting visual features and classifying them to labels readable for humans, and therefore the choice of this thesis. This section entails the architecture and technical details of the chosen pre-trained model Inception-v3, and the classification process in detail from image input to label output and result storage.

4.2.1 Inception-v3

Inception-v3 was presented in Szegedy et al. (2015), and is a deep CNN. The first version of Inception got its name from introducing a "network within a network" style implementation in the architecture. What this means is that in a layer of nodes, each node is itself a smaller neural network. The Inception architectures are iteratively developed, and what is new for Inception-v3 are a few discoveries on how to scale up deep CNNs without causing too much increase in computational complexity. In an iterative process, Szegedy et al. discover the usefulness of replacing larger convolutional filters with a network of smaller ones. An example is a filter of size $5 \times 5 = 25$ computations, can be replaced by a mini-network of two layers of 3×3 filters, ultimately yielding only $3 \times 3 + 3 \times 3 = 18$ computations. Eventually it was discovered that any $m \times m$ feature can be replaced by a multi-layer network of asymmetric $n \times 1$ filters. In practice, this works best for medium sized filters with m between 12 and 20, and dramatically reduces computational complexity. The resulting proposed architecture is a 42 layer deep network combining the various principles new to Inception-v3 and demonstrates 3.5% top-5 error and 17.3% top-1 error on the ILSVRC 2012 (Russakovsky et al., 2015) classification. The network was trained on the ImageNet dataset using the *stochastic gradient descent* (SGD) training algorithm.

[2] https://deeplearning4j.org/
[3] https://maven.apache.org/

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Figure 4.1: A screenshot of the table from Szegedy et al. (2015) detailing the Inception-v3 architecture outline.

Figure 4.1 shows the layer distribution of the proposed architecture. The next-to-last layer is a feature vector of 2048 dimensions, which is classified into 1000 categories using a softmax function. The softmax function is a generalization of binary logistic regression that allows a probability distribution over multiple classes.

Despite that many commonly tagged concepts are missing from the 1000 ImageNet categories, the precision to computational cost ratio of Inception-v3 makes it an ideal fit, especially considering the limitations of this thesis which inhibits training of a new model.

4.2.2 Classification

Here the method of obtaining the image classification predictions is detailed. The classification process starts with loading the pre-trained Inception-v3 model to a TensorFlow Graph object. Images are then converted to Tensors and executed on the graph one at a time. The images are processed through the network, and probabilities are distributed by the softmax function. The probabilities are sorted and the desired top n predictions are extracted. For practicality and simplicity this thesis will adopt *partial matching* in the evaluation stage as introduced by Li et al. (2015). This means that phrases are split up and treated as single words. Several of the categories in ImageNet are phrases, and therefore some light pre-processing has to be performed on the predictions. Hyphens and apostrophes are the only two signs occuring among the 1000 categories, and are simply replaced by white spaces. This preserves the semantic meaning of the words. For example, "lady's slipper" becomes "lady s slipper", and "table-tennis" becomes "table tennis". The single characters left by replacing apostrophes with white spaces are unproblematic, as they are disregarded by the preceding system components. After the light pre-processing, predictions are written to a file in local storage. In Figure 4.2 the reader can see a classified image along with its top 5 predictions. In Figure 4.3 is a diagram illustrating the image classification component.

Since this work does not deal with real-time processing, there are no specific requirements of computational speed. However, classifying an image using even a low budget

Figure 4.2: im1083 from MIRFLICKR with the top 5 predicted tags. "Electric guitar" is split into separate words.

Intel Core i5-5200U laptop CPU takes little more than half a second (even faster with an equal quality GPU). On Flickr, an average of 1.63 million photos were uploaded each day in 2017. This roughly equates to 19 every second. Thus, the classification speed equals around 10% of the photo stream even using a low budget laptop setup, which should be satisfying enough for real-time applications.

4.3 Word Embedding

As previously discussed in Chapter 2 and Chapter 3, there are two major challenges when predicting tags for user-generated content. Users' personal associations, incorrect tagging and the high degree of tag ambiguity lead to a high degree of noise. The other problem is the large conceptual gap between today's benchmark dataset ImageNet and users' wide interest span. As Izadinia et al. (2015) points out in their analysis almost half the most commonly used Flickr tags are indeed missing from the 1000 categories ImageNet vocabulary. This thesis will try to bridge some of this conceptual gap by utilizing word embedding models to expand the initial predictions from the classifier. Hopefully this method will help us predict tags and concepts far beyond the initial 1000 categories. Another advantage of using word embedding models in this way, is that when a tag correctly predicted from the classifier is used to extract word neighbors, chances are that all are semantically relevant if not an exact match. Training image classifiers of thousands of categories is close to impossible with poor computational setups, but the efficiency of Word2vec enables training of high-dimensional word embedding models from large amounts of text even on low budget hardware. This section details the word embedding models used for this approach, how they are trained, their strengths and possible weaknesses.

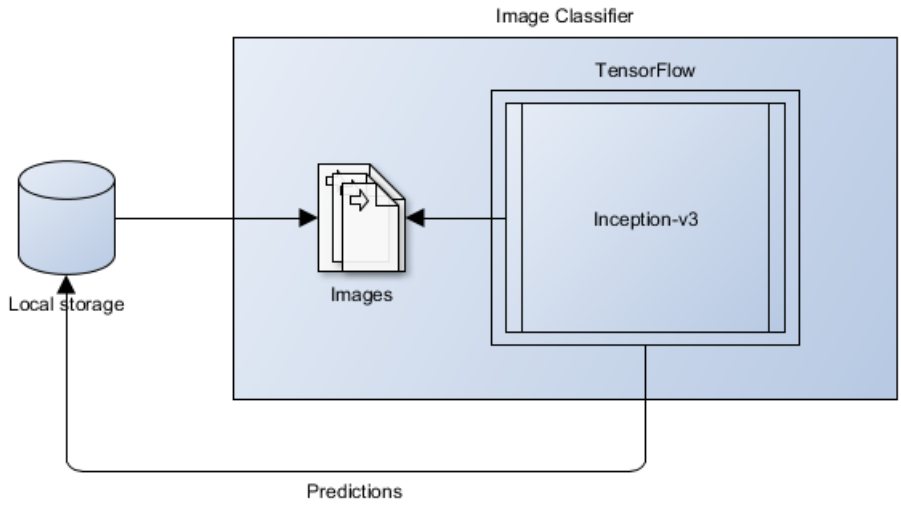

Figure 4.3: The image classification component.

4.3.1 Models

To help alleviate the problem of missing concepts, three different models are utilized in our approach. The first described is a pre-trained model, while the two others are trained for this thesis.

GoogleNews-SLIM

GoogleNews-vectors-negative300-SLIM[4], referred to as GoogleNews-SLIM in this thesis, is a narrowed down version of the enormous GoogleNews model trained by Mikolov et al. (2013b). The original GoogleNews model was trained on around 3 billion and had a vocabulary of 3 million words, making it 1.6 gigabytes (GB) in size. In models as large as these, loading and lookup times are slow. This is inconvenient for the work in this thesis, since a lookup has to be performed for each prediction token. The GoogleNews-SLIM however is downsized by crossing it with dictionaries; several english and one urban. The resulting 300-dimensional model has a vocabulary of 299,567 words in a 270 megabytes (MB) compressed Word2vec format, providing much faster loading and lookups. This model has a vastly rich vocabulary, to the point that it can almost serve as a dictionary. Because of this, retrieved neighboring words will almost always be semantically related to the input word, and will rarely contain nonsense. An example of this can be viewed in Table 4.2, where 8 of the 10 retrievals are indeed variations of the noun itself. This attribute can be both a strength and a weakness. On one hand the words will almost always be related. On the other, the number of categories (given the large vocabulary) may cause

[4]https://github.com/eyaler/word2vec-slim

it to be too specific and thus limit the span of concepts that can be reached when expanding predictions.

Flickr25K

To experiment, two models are trained for this thesis. The first is *Flickr25K*, a model trained on tags from the entire MIRFLICKR dataset. The tag files were iterated through to produce one large corpus of text, with the tags of each image treated as one sequence or sentence. The idea is that the model should be able to capture the relationships between tags, and learning what tags are used in concert. This way the model will hopefully find tags that are likely to appear together with the input word. The resulting text corpus consists of 220,872 words, with 63,779 of them unique. Training was accomplished using the skip-gram algorithm, as it perfectly matches the idea of predicting contextual tags. Minimum word frequency was set as 3 (words occuring less than 3 times are disgarded from vocabulary). A variety of configurations were tested for training and can be viewed in Table 4.1. The context window is denoted by CW in the table, while epochs are iterations over the whole corpus. Iter is the number of iterations over each sequence. Learning rate was set to 0.05 and the number of dimensions to 300 for all configurations.

Config	CW	Epochs	Iter
1	5	30	1
2	5	50	1
3	5	30	3
4	5	50	3
5	8	30	1
6	8	50	1
7	8	30	3
8	8	50	3

Table 4.1: Training configurations

The various models were evaluated on a small subset of MIRFLICKR not used in the final evaluation. The model using configuration 5 proved the most accurate, and was used for training the final model. Vocabulary of final model is 11,707 words, and with a size of only 62 MB. A 500-dimensional model was also attempted, but gave worse results and was discarded. Given the small size, both loading the model and queries are done extremely fast. Loading only takes 3-4 seconds, and queries can be done in 15 ms. In contrast, loading GoogleNews-SLIM takes about 25 seconds, and queries as much as 300 ms.

There are advantages to training the semantic space on text from a specific domain. Since Word2vec is great at learning context, the model may discover the syntactic relationship that tags have in the domain, which can be quite different from words in a normal text or news articles as the pre-trained model is trained on. That being said, the training data is made up of user-generated tags. The noise present in these are high because of ambiguities, individual associations, and incorrect tagging as discussed in Chapter 3. A few of the most irrational tags will be excluded given the minimum word frequency, but other are eventually encountered. The large variety and strange contexts of user tags is evident

in Table 4.2. An example is for the not-so-common word "seashore" which GoogleNews returns words like "coastline", "beachfront" and "boardwalk", while Flickr25K returns "hackspot", "wideaspect" and "seaside".

As this model was trained on tags from the validation dataset it is somewhat biased towards the data. However, note that fitting the model this way is actually highly comparable to training a model on for example recent data to discover and better predict trends. Therefore this bias should not in any way be discrediting towards the results.

Flickr368K

The second model, *Flickr368K*, is trained on a subset of the Flickr tags used by Li et al. (2015) for their word embedding training. With the whole corpus being a little too large for this thesis, a subset containing the first 368,575 sequences were chosen for training. As in the corpus used to train the first model, each sequence corresponds to the tags belonging to an image. The text contains 2,861,692 words and vocabulary size of the final model is 104,995 words. The same training configuration was used for this model, with 30 epochs, 8 word context window and a single iteration for each sequence. While this model is also trained on user-generated tags, the training corpus is much larger than Flickr25K and provides a vocabulary of almost 9 times the size. This increased number of categories should make it more specific than Flickr25K, still keeping the advantage of better representing user concepts than GoogleNews-SLIM. Loading the model takes around 15-20 seconds, and queries take approximately 100 ms.

Example Queries

To demonstrate the difference in vocabularies, Table 4.2 displays the retrieval of the 10 closest neighbors of 3 word queries for each of the word embedding models. "Coffee" and especially "sun" are commonly used words. "Seashore" however is relatively infrequent, something that becomes apparent in the retrievals.

4.3.2 Neighbor Vector Extraction

After loading a pre-trained model, several functions become available. The one of most interest to this thesis is for retrieving the n most similar words. Similarity of the vectors is computed by *cosine similarity*, meaning the cosine value of the angle between them in the semantic space. Cosine of 0° is 1, so words that are exactly alike will have cosine similarity of 1. This component of the system extracts the desired number of nearest words from each prediction generated by the image classification process. Predictions for each image is iterated over, and the word embedding model is queried with each prediction for finding the nearest words. Classification predictions of length 1 and 2 are ignored, since predicting tags of this length is of no particular interest. When querying models trained on huge corpora that is not user-generated, the nearest word is sometimes the word itself only with a capital first letter. These duplicates are removed. The predictions and n nearest words for each image are then merged and written to a new file in local storage. Figure 4.4 shows a diagram of the process.

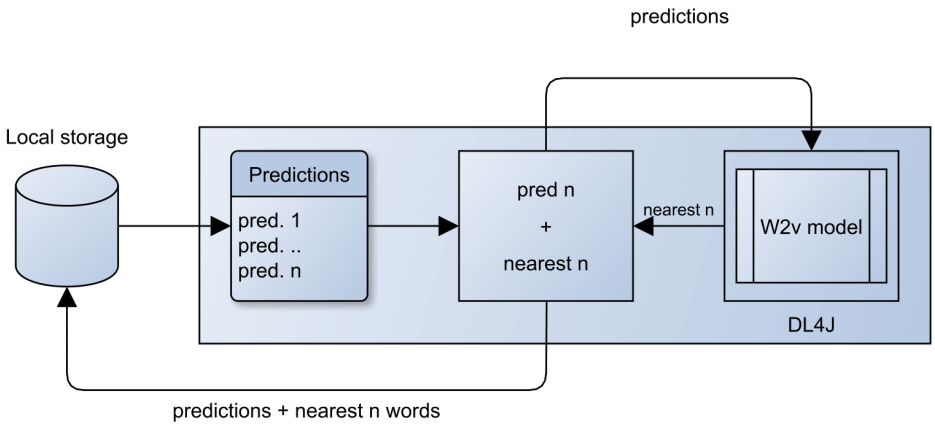

Figure 4.4: Nearest words extraction process.

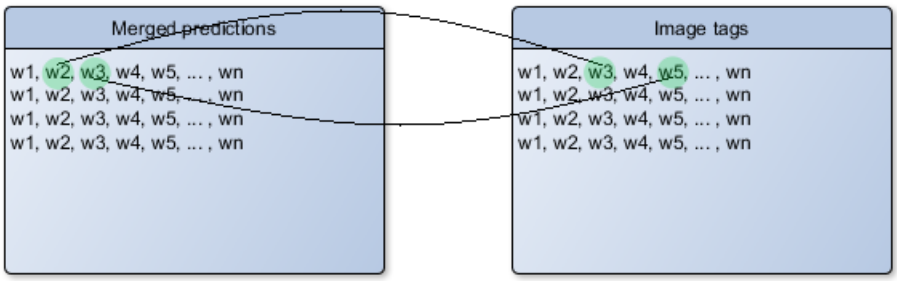

Figure 4.5: Illustration of the process of discovering predicted tags. Words highlighted in green indicate matches.

4.4 Tag Prediction Matching

The final process of the system is matching the augmented predictions and nearest words against the tags of the images they were generated from. As tags for every image are in separate files, the files are read from storage and written to a new file, with the tags of an image per line. Note that the tag files and image files are read from storage in the same order, and therefore get corresponding line numbers. Both the merged predictions file and the tag file are then read. Each word per line in the merged predictions file is evaluated against the line of tags to detect matching words. The resulting number of matches are successfully predicted tags. Figure 4.5 depicts the process.

Tag	Model	10 nearest words
coffee	GoogleNews-SLIM	coffees, Coffee, cappuccino, espresso, java, decaf, latte, Starbucks, expresso, smoothie
	Flickr25K	latte, breakfast, espresso, coffeecup, caffeine, coffeehouse, macchiato, nutella, latteart, cup
	Flickr368K	espresso, latte, cappucino, cafe, coffeeart, caffeine, coffeeshop, latteart, coffeecup, mocha
sun	GoogleNews-SLIM	sunlight, sunshine, rays, sunbeams, sunny, shade, suns, starlight, moonlight, sunburn
	Flickr25K	sunset, soleil, sol, sunrise, trees, clouds, sunbeam, silhouette, lakeerie, efsmmfusm
	Flickr368K	sunshine, sky, sunset, clouds, sunny, sunlight, cloud, sand, beach, shadow
seashore	GoogleNews-SLIM	beach, beaches, seashores, coastline, shoreline, beachfront, dunes, Seashore, coastal, boardwalk
	Flickr25K	hackspot, fadzly, wideaspect, fadzlymubin, shutterhack, seaside, facial, unprocessed, orang, beach
	Flickr368K	wildsingapore, shore, raffleslighthouse, betingbronok, intertidal, coastal, marinelife, echinodermata, bywydgwylltcymru, bydnatur

Table 4.2: Example queries of "coffee", "sun" and "seashore" showing 10 nearest words from each model.

Chapter 5

Experiments

This chapter describes the experiments of this thesis, meaning the details of how the proposed approach is evaluated. The dataset chosen for evaluating the tag prediction ability is discussed in the first half of the chapter. The second half entails details surrounding the automated evaluation, and describes the chosen performance measures.

5.1 Dataset

To perform evaluation of tag prediction or image annotation systems, access to a dataset is needed. If time is of an abundance, one can choose to gather sufficient amounts of data. This can sometimes be difficult, especially in regards to user-generated content. This content is often protected by strict licenses, and Social Media platforms such as Instagram have substantially limited the amount of data available through their application programming interface (API). Pre-made datasets can be found on the Web, although access to these may also be restricted and hard to gain. The dataset chosen for this thesis is MIRFLICKR25000[1] Huiskes and Lew (2008), partly because of its availability. It can be downloaded without any specific permission or access from their website.

MIRFLICKR comes in two sizes; 25000 and 1 million (1M). For this thesis, the MIRFLICKR25000 is best suited. It consists of 25000 images gathered from the Social Media platform Flickr, all under the Creative Commons license which is one of several public copyright licences that allows free distribution of contents. The set comes with several additions. As well as the user-generated tags, images are manually annotated according to concrete visual concepts. The last is exchangable image file format (EXIF) metadata, which describes things like camera information and settings, image information, time and location.

The creators have three special focuses in mind in regards to MIRFLICKR. The first is that it should be open and easily available. This they accomplish by having easy download access to both metadata and image files. They also want the dataset to be interesting. With

[1] http://press.liacs.nl/mirflickr/

this goal in mind, all images that are selected have a high rating of *interestingness*. Interestingness is a constantly changing metric used on Flickr for how interesting something is, and is composed by emphasizing on who (what users) comments, clicks and favorites the content. Practicality is the final characteristic they strive for, and is achieved by making metadata available in intuitively structured easy-to-access text files. Although primarily aimed at the image retrieval research community, these are very beneficial attributes for this thesis as well.

For the images of the dataset, the average number of tags is 8.94. This is a fair amount, and greater than the dataset used in Denton et al. (2015) (2.7), but fewer than in Park et al. (2016) (15.5). Denton et al. (2015) also restrict their evaluation subsets to include only images posted by English speaking countries, which should yield an increase in performance measures. No such restrictions will be enforced on the evaluation subset for this thesis. Most of the tags included are english, but there is a good amount of foreign as well. This will naturally cause a decrease in our performance measures to an unknown degree. Even though the average number of tags per image is decent for MIRFLICKR, the problem of the conceptual gap between ImageNet and user tags is evident here. Table 5.1 shows the distribution of the 20 most common tags in MIRFLICKR25000.

Tag	**Frequency**
explore	1483
sky	845
nikon	805
2007	794
blue	761
bw	737
canon	686
water	641
red	623
portrait	623
night	621
nature	596
sunset	585
green	569
clouds	558
macro	547
light	516
flower	510
abigfave	469
white	431

Table 5.1: The distribution of the 20 most common tags in MIRFLICKR25000.

Among these, none are represented as classification categories in ImageNet. The hope is still that the semantic and syntactic relationships learned by the word embedding models from this domain will enable us to predict some of these categories. Although none of the 20 most common tags are present, a number of subcategories are. For example, "flower"

itself is not a classification category, but "sunflower", "wallflower", "strawflower", "cornflower", "daisy" and more specific flower types are included.

User-generated tags are included in two forms in the dataset. One is a set of raw tags, and the other is "cleaned up" slightly by a pre-processing mechanism utilized by Flickr. In the pre-processing, the raw tags are converted to lower case, and phrases are joined together as one tag by removing white spaces. When matching predictions and tags, this thesis utilizes the pre-processed tags. Although phrases are likely to provide a few more matches, the amount is considered too insignificant in regards to the convenience of using the "cleaned up" tags.

5.2 Evaluation

Evaluating the art of tag prediction is both straightforward and difficult, depending on the context. The strict presence of accurately predicted tags can be easily measured and used to compute counting-based ratios. However, imagine the case of predicting tags for tag-based user profiling. Even if a prediction is not a perfect match for a tag, it may very well be semantically similar and still considered relevant. For example, it might belong to the same parent category. If the degree of relevance is high enough, it could be highly usable in applications such as content recommendation. With this thought in consideration, a short qualitative evaluation of the degree of relevance of predicted tags is performed in addition to the quantitative performance measures. This also adds a new dimension to evaluating the performance of the word embedding models. In the research field of user-generated tag prediction, performance measures are generally quite low, reinforcing the difficulty of the task.

The evaluation of the system of this thesis will be conducted on a subset consisting of 1000 MIRFLICKR images. This is considered a large enough amount to generate representative results.

5.2.1 Performance Measures

For quantitative evalution, performance measures common to the research field of tag prediction and image annotation are chosen. Precision, recall, F-measure and accuracy is computed for the whole evaluation subset.

To establish a result baseline, the first experiment is predicting tags using only the top five image classification predictions. However, a classification prediction is discarded from the five if the probability is below 1%. The reason for not using more than five of the image classification predictions is that the probability of it being correct heavily decreases after the first five. Several other configuration setups are chosen for the experiments. Given the novelty and difference of our approach, performance measures are not looked at for a specific number of tags like other similar works. This is not a problem, since nearly all works test compute their performances differently, and evaluate on different datasets. Most works produce a baseline which they compare their results to, and seem to emphasize these as much as those of other works. Because experiments and results in the research field vary this much, results in comparison to the chosen baseline are just as interesting as those of other works. This thesis evaluates combinations of top predictions and neighboring

words of different numbers. This will help determine what number of image classification predictions should be utilized in the nearest word extraction, and the amount of nearest words to retrieve per prediction. The following setups are used in the experiments:

- Top 1 prediction + 10 nearest words
- Top 2 predictions + 2 nearest words
- Top 2 predictions + 5 nearest words
- Top 3 predictions + 2 nearest words
- Top 3 predictions + 5 nearest words
- Top 5 predictions + 2 nearest words
- Top 5 predictions + 5 nearest words

The experiments are performed for each of the three word embedding models, and performance measures are computed for every setup. Measures are only computed for images with a non-zero set of tags.

Precision, or the positive predictive value, is the fraction of the number of true positives over the number of true positives and false positives. This metric can be said to signify the degree of quality of results. In the case of this thesis, precision equates to the number of correctly predicted tags over the total number of predicted tags. Let $Predictions(x_i)$ denote the set of predictions and nearest words predicted for the ith image x. $Tags(x_i)$ is the set of tags belonging to the ith image x. Let N denote the number of evaluation images with a non-zero set of tags. For the whole evaluation subset of N images this becomes:

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{|Predictions(x_i) \cap Tags(x_i)|}{|Predictions(x_i)|}$$

Recall is the fraction of true positives over true positives and false negatives among the images, and signifies completeness of results. For this thesis this equates to the number of correctly predicted tags over the number of tags belonging to the image, and is indicative of the ability to span the width of concepts in user-generated tags. For the whole evaluation subset of N images this becomes:

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\frac{|Predictions(x_i) \cap Tags(x_i)|}{|Tags(x_i)|}$$

F-score is the harmonic mean of precision and recall, and the trade-off between the two. F-score over the whole evaluation subset of N images is:

$$F = \frac{1}{N}\sum_{i=1}^{N}\frac{2(Precision(x_i) \times Recall(x_i))}{Precision(x_i) + Recall(x_i)}$$

Accuracy is the final metric computed, and is in this case defined as the fraction of images where at least one tag is correctly predicted. In contrast to precision, recall and F-score, accuracy is especially interesting since it is not affected by how many tags an image

has. For this reason, and that it indicates the number of matches achieved, it is arguably the most relevant measure to this thesis along with. Accuracy for an image x_i is given by:

$$Accuracy(x_i) = \begin{cases} 1 & \text{if } Predictions(x_i) \cap Tags(x_i) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Average accuracy over the whole evaluation subset is:

$$AverageAccuracy = \sum_{i=1}^{N} \frac{Accuracy(x_i)}{N}$$

Together these performance measures will be a help to better interpret results. Instead of focusing on a single metric, this combination allows better understanding of how the different setups affect the produced results.

Chapter 6

Results and Discussion

This chapter contains results from the experiments described in the previous chapter. They are presented in easy-to-read tables, and discussed in the preceding section. In the final section, a recapitulation is performed to give a better overview of key results.

6.1 Results

Since three separate word embedding models are evaluated in such a number of experiments, results for each model are presented separately. The results considered most significant are highlighted and summarized in the final section of the chapter.

6.1.1 GoogleNews-SLIM

Table 6.1 shows results of the pre-trained GoogleNews-SLIM word embedding model for the various configurations described.

Experiment	Precision	Recall	F-score	Accuracy
Baseline	**5.291**	1.720	**2.596**	12.93%
Top 1 + 10 nearest	1.927	2.331	2.110	14.96%
Top 2 + 2 nearest	3.064	1.876	2.327	14.28%
Top 2 + 5 nearest	2.095	2.338	2.209	16.53%
Top 3 + 2 nearest	2.797	2.031	2.353	15.41%
Top 3 + 5 nearest	1.916	2.465	2.156	17.32%
Top 5 + 2 nearest	2.643	2.249	2.430	17.09%
Top 5 + 5 nearest	1.824	**2.745**	2.192	**19.57**%

Table 6.1: Results for GoogleNews-SLIM in comparison to baseline.

6.1.2 Flickr25K

These are the results of the Flickr25K word embedding model trained for this thesis on the tags from the 25,000 images of MIRFLICKR. The following table shows the results in comparison to the baseline.

Experiment	Precision	Recall	F-score	Accuracy
Baseline	**5.291**	1.720	2.596	12.93%
Top 1 + 10 nearest	3.005	2.756	**2.876**	16.42%
Top 2 + 2 nearest	3.510	1.828	2.404	12.14%
Top 2 + 5 nearest	2.628	2.365	2.489	14.28%
Top 3 + 2 nearest	3.489	2.089	2.613	13.83%
Top 3 + 5 nearest	2.609	2.709	2.658	16.08%
Top 5 + 2 nearest	3.379	2.254	2.705	15.29%
Top 5 + 5 nearest	2.534	**2.961**	2.731	**18.33**%

Table 6.2: Results for Flickr25K in comparison to baseline.

6.1.3 Flickr368K

The Flickr368K results can be seen in Table 6.3, the word embedding model trained for this thesis on a subset of the tags used by Li et al. (2015) in their word embedding training. The subset consisted of tags from approximately 368,000 Flickr images.

Experiment	Precision	Recall	F-score	Accuracy
Baseline	**5.291**	1.720	2.596	12.93%
Top 1 + 10 nearest	2.547	2.966	2.741	17.43%
Top 2 + 2 nearest	3.895	2.479	3.030	16.64%
Top 2 + 5 nearest	2.638	3.126	2.861	18.78%
Top 3 + 2 nearest	3.668	2.718	3.122	18.33%
Top 3 + 5 nearest	2.482	3.448	2.886	20.92%
Top 5 + 2 nearest	3.509	3.008	**3.239**	20.80%
Top 5 + 5 nearest	2.347	**3.830**	2.911	**24.07**%

Table 6.3: Results for Flickr368K in comparison to baseline.

6.2 Discussion

6.2.1 Performance Measures

As mentioned in Section 5.1, the significant number of common tags in the dataset missing from ImageNet produces fairly low valued results in the baseline. Despite this, the precision of the baseline is the highest among the experiments, and surprisingly not that much lower than in Izadinia et al. (2015) (5.291 vs. 8.0), although they both train and evaluate

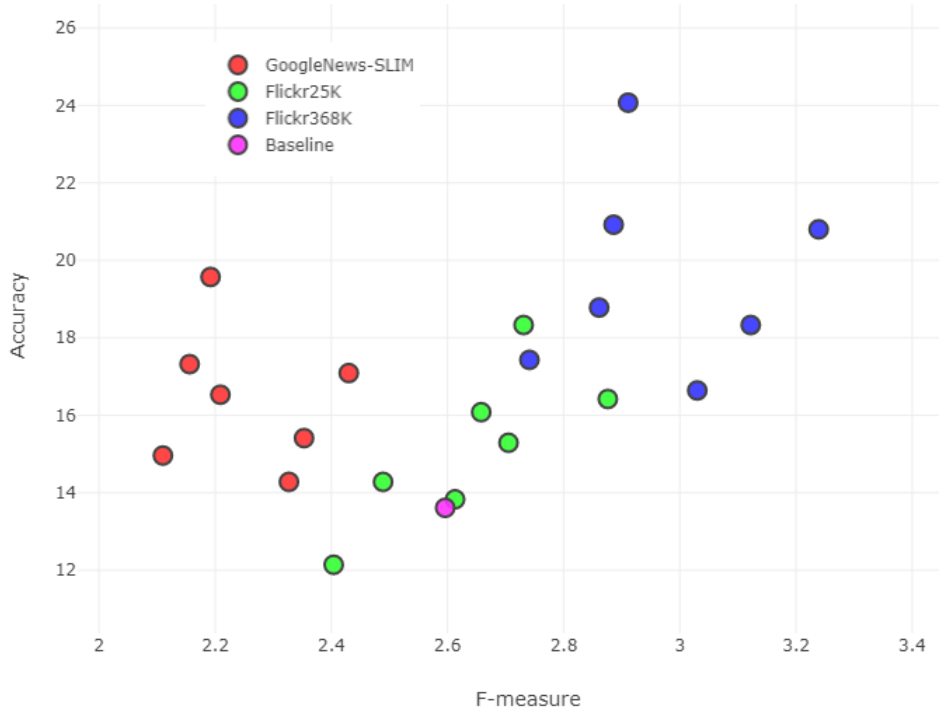

Figure 6.1: A graph displaying F-score to Accuracy. The F-score is displayed on the x-axis, and accuracy on the y-axis.

their CNN classifier on user-generated tags. The high precision in the baseline is caused by the low number of predictions (five) for that setup. The precision of the baseline would also likely drop drastically using an increased number of image classification predictions, since that 85% of the tags predicted by top five classifications are also predicted by the top one. The baseline also has a very low recall value, and confirms that the ImageNet categories are poorly represented in the common user concepts. The combination of high precision and very low recall results in a reasonable F-score, but taking the low accuracy into consideration as well diminishes its significance.

The distribution of the results are lower than some of the other works in the field. There are several reasons for this. In Park et al. (2016), the evaluation dataset include nearly two times the amount of average tags per image. This will make recall values slightly lower, but precision and especially accuracy increases. The most important reason for the differences however, is that Izadinia et al. (2015) and Garrigues et al. (2016) both train their classifiers on user-generated tags. Not only that, but they are trained on the most common tags as well, and restrict themselves to a lower number of categories unlike this approach. Results in Denton et al. (2015) are much more comparable to this thesis', but also restricts the number of prediction categories. What also becomes apparent by the results is that all setups except one (Top 2 + 2 nearest in Table 6.2) significantly increases accuracy of the tag prediction.

In Figure 6.1 is a two-dimensional graph displaying the results to provide a better visualization. What this graph and the results tell us is that the word embedding models all increase tag prediction ability. Of the three word embedding models, the pre-trained GoogleNews-SLIM performs the poorest. Although it has slightly higher and comparable accuracy to Flickr25K, the F-scores are notably lower. A reason for this could be that it is too specific, as discussed in Section 4.3.1. Another is that since it is not trained on words from the Flickr domain, it produces less relevant neighbors when a match is made for an image compared to the other two models. Flickr25K gives similar accuracy to GoogleNews-SLIM, but since it is trained on the domain, it seems to produce more relevant nearest words. This is even more evident by the results of the Flickr368K model. It provides the biggest increase in tag prediction ability, both in terms of F-score and accuracy. The most apparent reason to why this model performs better than Flickr25K is the size of the training data. While using the tags from 25,000 images in training results in a very small and tremendously fast model, it is unable to fully capture the semantics and syntax of the domain. The lesser amount of training data also increases risk of encountering out-of-vocabulary words, which results in an empty set of nearest words when the model is queried. Flickr368K is clearly the best performing model. It has the highest F-scores, and produces relevant nearest words to such a degree that it nearly doubles accuracy in the best performing setup, yielding an increase of just above 86%. Its precision is slightly higher than Flickr25K, but the recall values are on a different level and the main cause of the substantial increase in F-scores. In practice, this means that when matches are actually made, a bigger fraction of the set of tags belonging to the image is correctly predicted. The large increase in recall reveals that using a word embedding model trained on user-generated tags greatly helps in spanning the width of user concepts that were missing from the categories in the baseline. That this model is the best performer demonstrates the advantage to training on domain-specific text. It also shows that a sufficient amount of training data is necessary to properly capture syntactic and semantic relationships.

The experiments show that setups using the two nearest words mostly produce higher F-scores than the five nearest. This is due to that when the number of predictions grow, precision decreases. The opposite is true for recall, with the highest recall value (3.830) achieved by the top 5 + 5 nearest words setup using the Flickr368K model. An interesting observation is that the performance increase is substantial from the inclusion of two to five nearest words. This means that the the third, fourth and fifth nearest words are also very capable of producing correct matches, and displays the viability of utilizing more than just a couple of the nearest.

6.2.2 Relevance of Predicted Tags

To add a dimension to the evaluation, a few examples from the experiments are viewed to study the relevance of the predictions. As mentioned in Section 5.2, predicted tags can serve a purpose (e.g. content recommendation, personal photo search) despite not being an exact match, as long as they are semantically related to image contents. A prediction could arguably also be considered relevant if deemed as something that *could* be tagged in the image. The applications of the predicted tags are not necessarily restricted to the Social Media domain. Therefore, examples are chosen from experiments of both Flickr368K and GoogleNews-SLIM, as GoogleNews-SLIM generates predictions relevant for applications

Figure 6.2: im1807 from MIRFLICKR.

GoogleNews-SLIM	Flickr368K
alp	alp
volcano	volcano
mountain	**mountain**
tent	tent
valley	valley
crag	alps
pyrenean	berg
volcanoes	erupt
eruption	lava
mountains	mountains
mountainside	**peak**
tents	campsite
tented	tents
valleys	hills
foothills	

Table 6.4: Predicted tags for im1807 in Figure 6.2. Correctly predicted tags are highlighted.

beyond Flickr. The examples chosen are using the highest performing setup in terms of F-score, which is the top five image classification predictions and the two nearest words of each (top 5 + 2) in both models.

Figures 6.2, 6.3 and 6.4 show images from the evaluation subset. In Figure 6.2 the image contains a desert-looking mountaintop, and as evident by the predictions for the image in Table 6.4, both models produce tags that are arguably all to some degree related to the content. As often is the difference between these models, GoogleNews-SLIM finds multiple synonyms for "mountain" since it is more specific, while Flickr368K has a higher chance of producing another match from its nearest words. Most of the predictions generated are objects, and confirms the observation by Park et al. (2016)

Tag predictions for images from the two other examples seen in figures 6.3 and 6.4, have about the same degree of relevance and demonstrates that the overall quality of generated predictions are good. Nearly all tags are either conceptually related to the image contents, or deemed as possible tags that could be applied. A few exceptions can be seen in tables 6.5 and 6.6. "yellow" and "blue" appear in predictions for the image in Figure 6.3, and "dweeb", "gurl" and "colemonts" for the one in Figure 6.4. The latter example also shows a rare case of GoogleNews-SLIM outperforming the Flickr-trained model. In the few cases where this happens, the image often includes a high amount of tags that are objectively related to the image contents.

6.3 Summary

This chapter has reviewed and discussed the results generated by the experiments in this thesis. As the discussion contains an abundance of insights, keeping a clear overview can

Figure 6.3: im1894 from MIRFLICKR.

GoogleNews-SLIM	Flickr368K
red	**red**
wine	**wine**
bottle	bottle
goblet	goblet
yellow	yellow
blue	green
wines	redwine
chardonnay	merlot
bottles	bottles
jug	alcohol
goblets	**glass**
chalice	glassware

Table 6.5: Predicted tags for im1894 in Figure 6.3. Correctly predicted tags are highlighted.

Figure 6.4: im1606 from MIRFLICKR.

GoogleNews-SLIM	Flickr368K
running	running
shoe	**shoe**
barrel	**barrel**
loafer	loafer
jean	jean
sock	sock
ran	run
shoes	runners
footwear	shoes
crude	sneakers
dweeb	barrels
gurl	winery
denim	colemonts
jeans	gaultier
socks	knitting

Table 6.6: Predicted tags for im1606 in Figure 6.4. Correctly predicted tags are highlighted.

be difficult. Therefore, a recapitulation is included to highlight the most important points. These are presented in the following list:

1. All word embedding models provide a substantial increase in tag prediction ability from the baseline, thus proving them able to bridge part of the conceptual gap between user concepts and benchmark datasets for visual recognition challenges.

2. The models trained on user-generated Flickr tags give a higher increase in F-score than GoogleNews-SLIM. The domain-specific models are able to achieve a higher number of matches per image, and show that they better span the width of user concepts.

3. Domain-specific data serves as a better source for training when the goal is to maximize tag prediction ability.

4. The best performing experiment nearly doubles accuracy (12.93% vs. 24.07%) while still providing a higher F-score (2.596 vs. 2.911).

5. Overall quality of tag predictions are good, and can be utilized for other applications.

Chapter 7

Conclusion and Recommendations for Further Work

The findings of this thesis in relation to the research objectives are concluded in Section 7.1. Recommendations for further work and possible improvements to the approach are discussed in the final section.

7.1 Conclusion

Motivated by the benefits of utilizing knowledge extracted from the enormous amounts of user-generated content generated by Social Media, the goal of this thesis was to predict tags from the visual contents of images. The main contribution is a novel approach employing computer vision and word embedding models to perform image tag prediction for Social Media images. Image classification using convolutional neural networks was performed to extract the visual concepts and classify them to human labels, forming the basis of the predictions. Image classification models trained on todays benchmark datasets poorly represent the width of user concepts. To bridge this gap, word embedding models were utilized to retrieve semantically and syntactically related words of the image classification predictions, resulting in further predictions. The resulting model using the augmented predictions performs tag prediction with comparable accuracy to state-of-the-art.

In Section 1.3 a few objectives of the research were formalized as research questions:

Main RQ: *How to predict image tags from image visual content?*

RQ1: *How to predict tags using image classification?*

RQ2: *How to use word embedding models to help predict tags?*

RQ3: *What source of text is best for training the word embedding models?*

To examine how these objectives are accomplished, they are tied together with a few conclusions. After evaluating the approach and experiments described in this thesis, the following conclusions can be drawn:

- By using a CNN image classification model, visual features can be extracted and classified into labels readable by humans. This thesis showed that these labels can be used to predict tags. (**Main RQ & RQ1**)
- Utilizing word embedding models to retrieve additional semantically similar predictions increases tag prediction ability. (**Main RQ & RQ2**)
- Nearest words retrieved from word embedding models trained on domain-specific data provides the largest increase in tag prediction ability, and helps model the noise present in user-generated tags. (**Main RQ & RQ3**)
- Word embedding models can help bridge the conceptual gap between the ImageNet categories and user concepts, especially when trained on domain-specific data.
- Using image classification and word embedding together produces tag predictions of high relevance, and can be used in other applications.

7.1.1 Recommendations for Further Work

Although the approach of this thesis achieves its goal well, there are improvements to be made.

Improvements

The most protruding improvement that can be made to the approach of this thesis is training the image classifier on user-generated tags. This will likely achieve a significant increase in both precision and recall of image classification predictions. In addition, the image classification predictions will as a consequence provide a much better input for retrieving relevant nearest words from the word embedding model, since the input will be much more similar to the training data. Choosing a relevant subset of user concept to train for is difficult, but the most commonly used tags are a good place to start.

Another improvement that could increase prediction ability is implementing a separate classifier to choose a select number of tags from a set of candidate image classification predictions and their nearest words. In such a classification, user metadata such as gender, age, and friend relationships could be utilized to help select the proper tags. EXIF-metadata such as time and location could also be beneficial. The WordNet[1] hierarchy can also be exploited to extract parent words of the predictions to generate more candidate predictions.

[1] https://wordnet.princeton.edu/

Improving the performance of the image classifier is also a possibility, either by improving the architecture of the convolutional network or simply choosing a different pre-trained model. Experiments using more data or different sources of text for training the word embedding space might improve its performance as well.

Other applications

The approach could be used to generate tags for a personal photo collection to allow easier search. Since the predicted tags have a such high degree of relevance, a group of "hidden" tags could be applied to images and used for searching. This is a relevant application for both computers and smaller devices. For applications on smaller devices, a CNN architecture with a much lower computational cost could be employed such as MobileNet (Howard et al., 2017).

If utilized on a photo stream, the models provide an easy method of extracting visual information that can help determine trending topics and categories.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
URL `https://www.tensorflow.org/`

Baroni, M., Dinu, G., Kruszewski, G., 06 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors 1, 238–247.

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., Mar. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.
URL `http://dl.acm.org/citation.cfm?id=944919.944966`

Budura, A., Michel, S., Cudré-Mauroux, P., Aberer, K., 2009. Neighborhood-based tag prediction. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (Eds.), The Semantic Web: Research and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 608–622.

Cappé, O., Moulines, E., 06 2009. On-line expectation–maximization algorithm for latent data models 71, 593–613.

Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.-T., July 8-10, 2009. Nus-wide: A real-world web image database from national university of singapore. In: Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09). Santorini, Greece.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L., June 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255.

Denton, E., Weston, J., Paluri, M., Bourdev, L., Fergus, R., 2015. User conditional hashtag prediction for images. In: Proceedings of the 21th ACM SIGKDD International Con-

ference on Knowledge Discovery and Data Mining. KDD '15. ACM, New York, NY, USA, pp. 1731–1740.
URL http://doi.acm.org/10.1145/2783258.2788576

Duygulu, P., Barnard, K., Freitas, J. F. G. d., Forsyth, D. A., 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings of the 7th European Conference on Computer Vision-Part IV. ECCV '02. Springer-Verlag, London, UK, UK, pp. 97–112.
URL http://dl.acm.org/citation.cfm?id=645318.649254

Firan, C. S., Nejdl, W., Paiu, R., Oct 2007. The benefit of using tag-based profiles. In: 2007 Latin American Web Conference (LA-WEB 2007). pp. 32–41.

Garrigues, P., Farfade, S., Izadinia, H., Boakye, K., Kalantidis, Y., 2016. Tag prediction at flickr: a view from the darkroom. CoRR abs/1612.01922.
URL http://arxiv.org/abs/1612.01922

Godin, F., Slavkovikj, V., De Neve, W., Schrauwen, B., Van de Walle, R., 2013. Using topic models for twitter hashtag recommendation. In: Proceedings of the 22Nd International Conference on World Wide Web. WWW '13 Companion. ACM, New York, NY, USA, pp. 593–596.
URL http://doi.acm.org/10.1145/2487788.2488002

Goldberg, D., Nichols, D., Oki, B. M., Terry, D., Dec. 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM 35 (12), 61–70.

Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S., 2013. Deep convolutional ranking for multilabel image annotation. CoRR abs/1312.4894.
URL http://arxiv.org/abs/1312.4894

Grubinger, M., Clough, P., Müller, H., Deselaers, T., 2006. The iapr tc-12 benchmark – a new evaluation resource for visual information systems.

Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C., Sept 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 309–316.

Han, S., Pool, J., Tran, J., Dally, W. J., 2015. Learning both weights and connections for efficient neural networks. CoRR abs/1506.02626.
URL http://arxiv.org/abs/1506.02626

He, K., Sun, J., 2014. Convolutional neural networks at constrained time cost. CoRR abs/1412.1710.
URL http://arxiv.org/abs/1412.1710

He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR abs/1406.4729.
URL http://arxiv.org/abs/1406.4729

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. CoRR abs/1512.03385.
URL http://arxiv.org/abs/1512.03385

Hinton, G. E., Osindero, S., Teh, Y.-W., Jul. 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.
URL http://dx.doi.org/10.1162/neco.2006.18.7.1527

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861.
URL http://arxiv.org/abs/1704.04861

Huiskes, M. J., Lew, M. S., 2008. The mir flickr retrieval evaluation. In: MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval. ACM, New York, NY, USA.
URL http://press.liacs.nl/mirflickr/

Hung, C.-C., Huang, Y.-C., Hsu, J. Y.-j., Wu, D. K.-C., 2008. Tag-based user profiling for social media recommendation. In: Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at AAAI. pp. 49–55.

Izadinia, H., Russell, B. C., Farhadi, A., Hoffman, M. D., Hertzmann, A., 2015. Deep classifiers from image tags in the wild. In: Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions. MMCommons '15. ACM, New York, NY, USA, pp. 13–18.
URL http://doi.acm.org/10.1145/2814815.2814821

Karayev, S., Hertzmann, A., Winnemoeller, H., Agarwala, A., Darrell, T., 2013. Recognizing image style. CoRR abs/1311.3715.
URL http://arxiv.org/abs/1311.3715

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., Fei-Fei, L., 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.
URL https://arxiv.org/abs/1602.07332

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., USA, pp. 1097–1105.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., Nov 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86 (11), 2278–2324.

Li, X., Liao, S., Lan, W., Du, X., Yang, G., 2015. Zero-shot image tagging by hierarchical semantic embedding. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '15. ACM, New York, NY, USA, pp. 879–882.
URL http://doi.acm.org/10.1145/2766462.2767773

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: common objects in context. CoRR abs/1405.0312.
URL http://arxiv.org/abs/1405.0312

Makadia, A., Pavlovic, V., Kumar, S., 2008. A new baseline for image annotation. In: Proceedings of the 10th European Conference on Computer Vision: Part III. ECCV '08. Springer-Verlag, Berlin, Heidelberg, pp. 316–329.
URL http://dx.doi.org/10.1007/978-3-540-88690-7_24

McCulloch, W. S., Pitts, W., Dec 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5 (4), 115–133.
URL https://doi.org/10.1007/BF02478259

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.
URL http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. CoRR abs/1310.4546.
URL http://arxiv.org/abs/1310.4546

Monay, F., Gatica-Perez, D., 2003. On image auto-annotation with latent space models. In: Proceedings of the Eleventh ACM International Conference on Multimedia. MULTIMEDIA '03. ACM, New York, NY, USA, pp. 275–278.
URL http://doi.acm.org/10.1145/957013.957070

Monay, F., Gatica-Perez, D., 2004. Plsa-based image auto-annotation: Constraining the latent space. In: Proceedings of the 12th Annual ACM International Conference on Multimedia. MULTIMEDIA '04. ACM, New York, NY, USA, pp. 348–351.
URL http://doi.acm.org/10.1145/1027527.1027608

Murray, N., Marchesotti, L., Perronnin, F., June 2012. Ava: A large-scale database for aesthetic visual analysis. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2408–2415.

Murthy, V. N., Maji, S., Manmatha, R., 2015. Automatic image annotation using deep learning representations. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. ICMR '15. ACM, New York, NY, USA, pp. 603–606.
URL http://doi.acm.org/10.1145/2671188.2749391

Nov, O., Naaman, M., Ye, C., 2008. What drives content tagging: The case of photos on flickr. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '08. ACM, New York, NY, USA, pp. 1097–1100.
URL http://doi.acm.org/10.1145/1357054.1357225

Park, M., Li, H., Kim, J., 2016. HARRISON: A benchmark on hashtag recommendation for real-world images in social networks. CoRR abs/1605.05054.
URL http://arxiv.org/abs/1605.05054

Patterson, G., Xu, C., Su, H., Hays, J., 2013. The sun attribute database: Beyond categories for deeper scene understanding. International Journal of Computer Vision 108, 59–81.

Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A., 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. CoRR abs/1603.05279.
URL http://arxiv.org/abs/1603.05279

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65–386.

Ruocco, M., Ramampiaro, H., 2015. Geo-temporal distribution of tag terms for event-related image retrieval. CoRR abs/1504.07350.
URL http://arxiv.org/abs/1504.07350

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115 (3), 211–252.

Schoen, H., Gayo-Avello, D., Metaxas, P. T., Mustafaraj, E., Strohmaier, M., Gloor, P., 2013. The power of prediction with social media. Internet Research 23 (5), 528–543.
URL https://doi.org/10.1108/IntR-06-2013-0115

Sermanet, P., Chintala, S., LeCun, Y., 2012. Convolutional neural networks applied to house numbers digit classification. CoRR abs/1204.3968.
URL http://arxiv.org/abs/1204.3968

Sermanet, P., LeCun, Y., July 2011. Traffic sign recognition with multi-scale convolutional networks. In: The 2011 International Joint Conference on Neural Networks. pp. 2809–2813.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556.
URL http://arxiv.org/abs/1409.1556

Strohmaier, M., Körner, C., Kern, R., 2010. Why do users tag? detecting users' motivation for tagging in social tagging systems.
URL https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1497

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2014. Going deeper with convolutions. CoRR abs/1409.4842.
URL http://arxiv.org/abs/1409.4842

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. CoRR abs/1512.00567.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L., 2015. The new data and new challenges in multimedia research. CoRR abs/1503.01817.
URL http://arxiv.org/abs/1503.01817

Tkachenko, N., Jarvis, S., Procter, R., 02 2017. Predicting floods with flickr tags. PLOS ONE 12 (2), 1–13.
URL https://doi.org/10.1371/journal.pone.0172870

von Ahn, L., Dabbish, L., 2004. Labeling images with a computer game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '04. ACM, New York, NY, USA, pp. 319–326.
URL http://doi.acm.org/10.1145/985692.985733

Werbos, P. J., 01 1974. Beyond regression : new tools for prediction and analysis in the behavioral sciences /.

Weston, J., Bengio, S., Usunier, N., 2011. Wsabie: Scaling up to large vocabulary image annotation. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI.

Weston, J., Chopra, S., Adams, K., 2014. #tagspace: Semantic embeddings from hashtags. In: EMNLP.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., Torralba, A., June 2010. Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3485–3492.

Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X., June 2015. Learning from massive noisy labeled data for image classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2691–2699.

Yin, D., Xue, Z., Hong, L., Davison, B. D., 2010. A probabilistic model for personalized tag prediction. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '10. ACM, New York, NY, USA, pp. 959–968.
URL http://doi.acm.org/10.1145/1835804.1835925

Zhang, H., Berg, A. C., Maire, M., Malik, J., 2006. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2. CVPR '06. IEEE Computer Society, Washington, DC, USA, pp. 2126–2136.
URL http://dx.doi.org/10.1109/CVPR.2006.301

Zhang, L., Tang, J., Zhang, M., 2012. Integrating temporal usage pattern into personalized tag prediction. In: Sheng, Q. Z., Wang, G., Jensen, C. S., Xu, G. (Eds.), Web Technologies and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 354–365.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A., 2014. Learning deep features for scene recognition using places database. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1. NIPS'14. MIT Press, Cambridge, MA, USA, pp. 487–495.
URL http://dl.acm.org/citation.cfm?id=2968826.2968881