



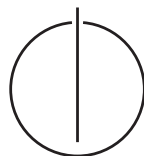
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

**Transfer- and Multitask Learning for  
aspect-based Sentiment Analysis using  
Google Transformer Architecture**

Felix Schober





DEPARTMENT OF INFORMATICS

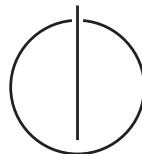
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

**Transfer- and Multitask Learning for  
aspect-based Sentiment Analysis using  
Google Transformer Architecture**

**Transfer- und Multitask Learning für  
aspektbasierte Sentimentanalyse mit der  
Google Transformer Architektur**

Author:	Felix Schober
Supervisor:	PD Dr. Georg Groh
Advisor:	Gerhard Hagerer M.Sc.
Submission Date:	15.05.2019



I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 15.05.2019

Felix Schober

## Acknowledgments

# Abstract

# Contents

<b>Acknowledgments</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Outline . . . . .	1
<b>2 Related Work</b>	<b>2</b>
<b>3 Theoretical Background</b>	<b>3</b>
3.1 Methodology . . . . .	3
3.1.1 Performance Measurements . . . . .	3
3.1.2 Cross Validation . . . . .	4
<b>4 Method</b>	<b>6</b>
<b>5 Experimental Setup</b>	<b>7</b>
5.1 Data . . . . .	7
5.1.1 Conll-2003 - Named Entity Recognition . . . . .	7
5.1.2 SemEval-2016 - Restaurants and Laptops . . . . .	7
5.1.3 GermEval-2017 - Deutsche Bahn Tweets . . . . .	7
5.1.4 Organic-2019 - Organic Comments . . . . .	7
<b>6 Discussion of Results</b>	<b>8</b>
6.1 Hyper Parameter Optimization . . . . .	8
6.2 Results for Named Entity Recognition . . . . .	8
6.3 Results for Aspect-Based Sentiment Analysis . . . . .	8
6.3.1 GermEval-2017 . . . . .	8
6.3.2 GermEval-2017 . . . . .	8
6.3.3 Organic-2019 . . . . .	8
<b>7 Conclusion</b>	<b>9</b>
7.1 Future Work . . . . .	9

## *Contents*

---

<b>Acronyms</b>	<b>10</b>
<b>List of Figures</b>	<b>13</b>
<b>List of Tables</b>	<b>14</b>

# **1 Introduction**

## **1.1 Motivation**

## **1.2 Outline**



## 2 Related Work

## 3 Theoretical Background

### 3.1 Methodology

#### 3.1.1 Performance Measurements

##### Precision - Recall

The most used measure for the precision of food classifiers is the average accuracy which is calculated by dividing the number of correct matches and the total number of samples. Accuracy, however, gives no information about the underlying conditions. It is a measure of overall performance. To have a higher chance of suggesting the correct items, future systems may present a list of options that the user can choose from. Intuitively, the accuracy is much higher if a classifier can present a list of items with high confidences instead of only one item because the problem is much easier. Accuracy, however, does not measure how easy a problem is. If a classifier were able to suggest all classes as options the accuracy would always be 100% although the results are not useful at all.

The combination of precision and recall objectively measures the actual relevance and performance of a classifier for a class of images because it includes the amount of considered items and the correct predictions. In this case the amount of considered items changes based on how many items the classifier can suggest. Precision and recall is defined as:

$$Precision = \frac{T_p}{T_p + F_p} \quad Recall = \frac{T_p}{T_p + F_n}. \quad (3.1)$$

- True positives  $T_p$  is the number of correctly classified images of a class.
- False positives  $F_p$  are all images that the classifier predicted to be positive but are in reality negative. (Type I Error)
- False negatives  $F_n$  are all images that are positive (belong to the class) but are labeled as negative (do not belong to class) (Type II Error)

A high recall means that many images were matched correctly and a high precision denotes a low number of incorrectly classified images. The bigger the area under the Precision-Recall curve the better the classifier.

### Null Error Rate

The null error rate is a baseline for any classification task that calculates the accuracy if a classifier would just predict the class with the most images.

### Confusion Matrix

Confusion matrices are one of the most important metrics to understand why a classifier struggles with certain classes while getting a high precision with others. As the name suggests, a confusion matrix tells if the classifier "confuses" two classes.

A confusion matrix for  $n$  classes is always a  $n \times n$  matrix where columns represent the actual images classes and rows represent the predicted image classes so if the diagonal of the matrix has high values this means that the classifier makes correct predictions.

### Categorical Cross-Entropy

The categorical cross-entropy  $L_i$  is an error function that is used for the training of neural networks in classification tasks as the objective function. It is more versatile than the accuracy or the Mean Squared Error (MSE) because it takes the deviations of the predicted label  $p_{i,j}$  and the actual label  $t_{i,j}$  into account and weights the "closeness" of the prediction with the logarithm. For classification, cross entropy is more useful than MSE because MSE gives too much emphasis on incorrect predictions. The categorical cross entropy function is defined as:

$$L_i = - \sum_j t_{i,j} \log(p_{i,j}) \quad (3.2)$$

The loss values that are used for the discussion of results for neural networks are the average values of the categorical cross-entropy (Average Cross-Entropy Error (ACE)).

#### 3.1.2 Cross Validation

Cross validation is one of the most essential techniques to evaluate real-world classification performance. Classifiers like Support Vector Machines (SVMs) or neural networks are always better on data they have already seen. This is called overfitting (see section ??). By training and testing on the same data the classification performance would be much better than the actual real world performance. To test if a classifier can actually

work with samples it has not seen cross validation divides the dataset into different partitions.

For most tasks it is sufficient to divide the dataset into a training and a test set. The data in the training set is used to train the classifier and the test data is used to evaluate it with data it has not seen before.

#### **k-fold Cross Validation**

To make the classification evaluation even more robust,  $k$ -fold cross validation is used. By applying  $k$ -fold cross validation the dataset is randomly partitioned into  $k$  different parts.  $k - 1$  parts are used for training and one part is used for the evaluation. This process is repeated  $k$ -times and after each iteration the parts are exchanged so that at the end, each sample was used for training and for validation. Calculating the mean of the  $k$  evaluations gives a much more robust measurement because the evaluation does not depend on the difficulty of the test partitions.

## 4 Method

# **5 Experimental Setup**

## **5.1 Data**

**5.1.1 Conll-2003 - Named Entity Recognition**

**5.1.2 SemEval-2016 - Restaurants and Laptops**

**5.1.3 GermEval-2017 - Deutsche Bahn Tweets**

**5.1.4 Organic-2019 - Organic Comments**

## **6 Discussion of Results**

### **6.1 Hyper Parameter Optimization**

### **6.2 Results for Named Entity Recognition**

### **6.3 Results for Aspect-Based Sentiment Analysis**

#### **6.3.1 GermEval-2017**

#### **6.3.2 GermEval-2017**

#### **6.3.3 Organic-2019**

## **7 Conclusion**

### **7.1 Future Work**



# Acronyms

**ACE** Average Cross-Entropy Error.

**API** Application Programming Interface.

**ATM** Amazon Mechanical Turk.

**BoW** Bag of Words.

**BRIEF** Binary Robust Independent Elementary Features.

**CenSurE** Center Surround Extrema.

**CIFAR** Canadian Institute for Advanced Research.

**CNN** Convolutional Neural Network.

**CPU** Central Processing Unit.

**CRF** Conditional Random Field.

**csv** Comma Separated Values.

**CUDA** Compute Unified Device Architecture.

**DoG** Difference of Gaussians.

**ETHZ** Eidgenössische Technische Hochschule Zürich.

**FAST** Features from Accelerated Segment Test.

**FRIDa** FoodCast Research Image Database.

**GB** Giga Bytes.

**GPS** Global Positioning System.

**GPU** Graphics Processing Unit.

**HOG** Histogram of Oriented Gradients.

**HSV** Hue, Saturation, Value.

**HTML** Hypertext Markup Language.

**hyponym** children of a synset.

**ILSVRC** ImageNet Large Scale Visual Recognition Challenge.

**IO** Input / Output.

**KNN** K-nearest Neighbors.

**LBP** Local Binary Patterns.

**LoG** Laplacian of Gaussian.

**MSE** Mean Squared Error.

**NRLBP** Non Redundant Local binary patterns.

**ORB** Oriented FAST and Rotated BRIEF.

**PFID** Pittsburgh Fast Food Image Dataset.

**PHOW** Pyramid Histogram of Visual Words.

**PRICoLBP** Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern.

**RAM** Random Access Memory.

**RBF** Radial Basis Function.

**RGB** Red Green Blue.

**SIFT** Scale-invariant Feature Transform.

**std** Standard Deviation.

**SURF** Speeded up Robust Features.

**SVM** Support vector machine.

**synset** Synonym Set.

**TADA** Technology Assisted Dietary Assessment.

**TFC** TUM Food Cam.

**TUM** Technische Universität München.

**UEC** University of Electro Communications.

**UPMC** Université Pierre et Marie Curie.

**URL** Uniform Resource Locator.

**US** United States.

**wnid** WordNet ID.

**ZCA** Zero-Phase Component Analysis.

## List of Figures

## List of Tables