

Using Transfer Learning Effectively: A Characterization of Negative Transfer in Data and Ways to Avoid it

Degree program:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's program
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	Mikael Engels
Identification number:	851842797
Date:	28 Jun 2018
Thesis supervisor	Dr. Stefano Bromuri
Second reader	Dr. Arjen Hommersom
Version number:	40
Status:	Final version

Abstract

This thesis focuses on the effectiveness of transfer learning and how negative transfer could be avoided within machine learning. Transfer learning is used to transfer knowledge between source and target datasets and can, if used correctly, have significant advantages for businesses. Situations where transfer learning could be useful; if the target data is of a poor quality, when there is no time to train a model or if there minimal training data available.

This thesis created several models that learn from a source (a dataset with text reviews and the corresponding sentiment) and applies the knowledge to all kinds of new “cheap” and “easy to get” unlabeled target datasets which leads to significantly reduction of the costs of labelling data.

However, transfer learning does not always work well and could even lead to negative transfer; this is where the transfer learning is no longer effective. This research showed that the transfer learning variant “Multiple Source Transfer Learning” works the best with review data. Secondly a similar word distribution in the dataset, lead to more positive transfers and prevent the negative transfer from happening. The similarity could be measured with WordNet or comparing dataset metrics.

Keywords

Transfer Learning, Negative Transfer, Machine Learning, Data Science Management, Text Mining, Deep Learning

Summary

This thesis focuses on the effectiveness of transfer learning and how the negative transfer could be avoided. Transfer learning is a subfield of machine learning, and like the name suggests, it is about transferring learning or knowledge from source to target datasets. Transfer learning can have, if used correctly, significant advantages for businesses.

Transfer learning could help situations where the target data has a poor quality, when there is no time to train the model, or when limited training data is available to train the model. Real business situation where transfer learning could be used are; recognize future customer sentiment based on actual collected customer communication, self-driving cars predicting traffic situations learned from a source simulator dataset without having multiple crashes itself or recognize tumor cells with pre-trained knowledge.

First this thesis contains a theoretical framework to find out the concepts of transfer learning and secondly an experiment with multiple transfer learning approaches, to find out which technique, datasets and algorithms performs the best. This has given insights when negative transfer happens and how it could be prevented. CRISP-DM is used for the structure, which is an industry standard for data analytic projects.

For this research is worked with review datasets from mainly Amazon and Yelp. Customers gave their opinion about products and services by writing in text and give star-ratings. The models developed in this thesis learn from review data and transferred this to target datasets to predict the star-rating based on the text. Because all the review datasets are unbalanced, on average 78% positive class (4 or 5-star ratings), first are the reviews preprocessed and then resampled. In this thesis a combination of SMOTE oversampling and undersampling used to rebalance the data. After this, the reviews will be tokenized, chopped in words and put in a word-vector by TF-IDF. The vector gets processed by Latent Semantic Index, to reduce the complexity from an enormous number of features to the number of dimensions, also known as topics. This is getting processed by an algorithm, and the knowledge is getting then transferred (applied) to the other dataset.

The goal is not to create a model with the highest accuracy, but to do better than the non-transfer machine learning model; this is called a positive transfer. In the experiment it was found that a Transductive-Transfer-Learning model (fully trained on the source) worked better but not as good as the Inductive-Transfer-Learning also known as Multiple Source Transfer Learning models (trained on source and 70% of the target data) work the best with 45% of positive transfers, 33% had no improvement and 22% of negative transfers compared to the machine learning model. This is significant proof that in this experiment transfer learning was successful.

The effect of the transfers and the prevention of negative transfer improves when the right TL technique is chosen and the distribution of the words between the datasets is more similar, for instance, Amazon Health & Personal Care products and Amazon Beauty products are more similar than Yelp restaurant reviews about services or Amazon Automotive products. This similarity could be measured with WordNet and compare the dataset metrics, and this thesis provides a way to do this. It also helps to create a model that reuses data from the sources to improve the classification. So is presented a model that uses 100% of the source, and 70% of the target to train from and the classification could be further improved by applying Topic Modelling with Latent Semantic Indexing to reduce the vector complexity.

Contents

Abstract	ii
Keywords	ii
Summary.....	iii
Contents	iv
1. Foreword.....	1
1.1. Introduction	1
1.2. Exploration of the topic.....	1
1.3. Motivation and relevance	2
1.4. Problem statement	2
1.5. Terms of reference.....	2
1.6. Main lines of approach.....	3
2. Theoretical framework	4
2.1. Research approach.....	4
2.2. Implementation	5
2.2.1. High impact papers.....	6
2.2.2. Medium impact papers	7
2.2.3. Low impact papers	8
2.2.4. Sources quality	9
2.3. Results and conclusions	9
2.3.1. Q1. What is Transfer Learning?	9
2.3.2. Q2. What is a Negative Transfer?.....	10
2.3.3. Q3. Which disciplines make use of Transfer Learning?	11
2.3.4. Q4. What does TL mean in computer science?	12
2.3.5. Q5. What approaches are often used in TL?	13
3. Methodology.....	16
3.1. Conceptual design: select the research method(s).....	16
3.2. Technical design: Elaboration of the method	16
3.2.1. Focus.....	16
3.2.2. Tools	17
3.2.3. Measuring.....	17
3.3. Reflection validity, reliability and ethical aspects	18
4. Results.....	19
4.1. CRISP-DM: Business Understanding.....	19
4.1.1. Determine Business Objective.....	19
4.1.2. Asses Situation	19
4.1.3. Datasets.....	19
4.2. CRISP-DM: Data Understanding.....	20

4.2.1.	Describe and collect initial data	20
4.2.2.	Explore data.....	21
4.2.3.	WordNet: Most commonly used words	23
4.2.4.	Class imbalance	25
4.2.5.	Missing values	27
4.2.6.	Correlation matrix	28
4.3.	CRISP-DM: Data Preparation	28
4.3.1.	Clean data.....	28
4.3.2.	Construct & integrate data	29
4.3.3.	Further cleaning	30
4.4.	CRISP-DM: Modelling	30
4.4.1.	Selected techniques	30
4.4.2.	Find optimal values & operators	32
4.4.3.	SMOTE & Balancing classes	32
4.4.4.	Build models	33
4.4.1.	Asses models	34
4.5.	CRISP-DM: Evaluation	35
4.5.1.	Target=HPC.....	35
4.5.2.	Target=Beauty	37
4.5.3.	Target=Automotive	39
4.5.4.	Target=Yelp	42
4.5.5.	Evaluate results	44
4.5.1.	Extra experiment 1 – Deep Learning & Neural Nets	47
4.5.2.	Extra experiment 2 – Add extra datasets	49
5.	Conclusion, discussion, and recommendations, reflection	53
5.1.	Conclusion.....	53
5.1.1.	Q6. When is Transfer Learning most effective?	53
5.1.2.	Q7. How to avoid Negative Transfer?	53
5.1.3.	Main research question.....	53
5.2.	Discussion.....	54
5.3.	Practice recommendations	55
5.4.	Recommendations for further research	55
5.5.	Reflection	55
6.	References	56
	Appendix 1; Explanation of all used operators.....	58

1. Foreword

1.1. Introduction

This thesis is part of the master's study "Business Process Management & IT" of the Open University of the Netherlands. A data-analytic-thesis was written about:

"Using Transfer Learning Effectively: A Characterization of Negative Transfer in Data and Ways to Avoid it."

Transfer Learning (TL) is a subfield of Machine Learning (ML) and belongs to Artificial Intelligence. TL is used for transferring knowledge between datasets. It gives smaller datasets the possibility to use the knowledge gathered from other datasets, saves the time of collecting its knowledge and creates a higher classification accuracy. Despite this simple concept, TL can be challenging in practice. Sometimes TL leads to Negative Transfer (NT), when the transfer of knowledge is less efficient than building the knowledge from the beginning (S. J. Pan & Yang, 2010). Therefore, this research aimed to examine 1) the difficulties with TL, 2) how TL could work better and 3) how to avoid NT.

1.2. Exploration of the topic

The amount of data produced by computer systems today is enormous, have a high variety and increasing fast. These variables volume, variety, and speed distinguish standard data from 'Big Data' (Gandomi & Haider, 2015). Analyzing Big Data manually would be tedious and time-consuming, and in most cases unviable. New technologies like distributed computing, cloud, and increasing processing power, make it possible to analyze and process these enormous amounts of data (Helms, 2015).

Businesses are constantly trying to improve their products and services by using data. Moreover, data is becoming more and more important to support various innovations (Helms, 2015). In the field of data analytics, also referred as data mining, various algorithms try to 'mine' data for valuable information (Boisot & Canals, 2004). In ML, is tried to automate an algorithm, for instance, to 'learn' automatically from a dataset. Some researchers describe TL as transferring knowledge from the source to target task (Torrey & Shavlik, 2009).

People can re-use knowledge from previous tasks and apply them to new tasks (Woodworth & Thorndike, 1901). For example, people learned to write computer code A, can easily apply this knowledge when learning computer code B. ML algorithms work differently. Traditional ML algorithms must learn every task from the beginning every time they get a new task. They 'forget' what they have learned from the previous dataset and need to acquire knowledge from the beginning. Thus, TL finds a way to acquire and store the knowledge from the source task and re-apply this to the target task (Weiss, Khoshgoftaar, & Wang, 2016).

The early principles of TL date back to the 20th century (Woodworth & Thorndike, 1901). During this time, scientists used different definitions for TL, like lifelong learning, knowledge transfer, and meta-learning (Lu et al., 2015). TL is a hot topic in ML, and not without a reason. Using knowledge from specific domains can facilitate predictive modeling in other domains (Lu et al., 2015). Amongst other cases, the usefulness thereof is shown in a recent case study on ImageNet (Krizhevsky, Sutskever, & Hinton, 2012). Where the knowledge of 1.2 million classified pictures, was used to improve the

classification of recognizing tumour cells successfully (Ehteshami Bejnordi, Veta, Johannes van Diest, & et al., 2017).

1.3. Motivation and relevance

From a business perspective is it useful to learn about effective strategies to apply TL and to avoid NT since it could lead to improved efficiency and reduced costs (Egan, Yang, & Bartlett, 2004). TL could give algorithms an intelligence boost and let algorithms learn from one dataset to the other. When done properly and without NT, TL should optimize the algorithms and lead to better quality and performance classification.

From a research perspective, there are many advantages of effective TL and avoiding NT. When algorithms can re-use knowledge, this will bring opportunities beyond our imagination for new research, such as automatically improving algorithms. More than 700 academic papers support the relevance of TL in the last five years, these are a definite proof that there is a need for TL (Weiss et al., 2016). However, even though there is much research on TL, not much is known about how to use it effectively.

1.4. Problem statement

Literature suggests that there is lots of room for improvement in TL (Weiss et al., 2016). There is much research about TL, but not much that focuses on NTs (Lu et al., 2015). This is surprising since the literature suggests that NT correlate strongly with for bad results in TL tasks (Torrey & Shavlik, 2009; Weiss et al., 2016).

1.5. Terms of reference

The central research question derived from the problem statement is:

<i>How to use Transfer Learning effectively and which factors cause a Negative Transfer?</i>
--

To answer the central question, the following sub-questions were made:

Q1. What is Transfer Learning?
Q2. What is a Negative Transfer?
Q3. Which disciplines make use of Transfer Learning?
Q4. What does Transfer Learning mean in computer science?
Q5. What approaches are often used in Transfer Learning?
Q6. When is Transfer Learning most effective?
Q7. How to avoid Negative Transfer?

1.6. Main lines of approach

Below, the structure of the research (Figure 1).

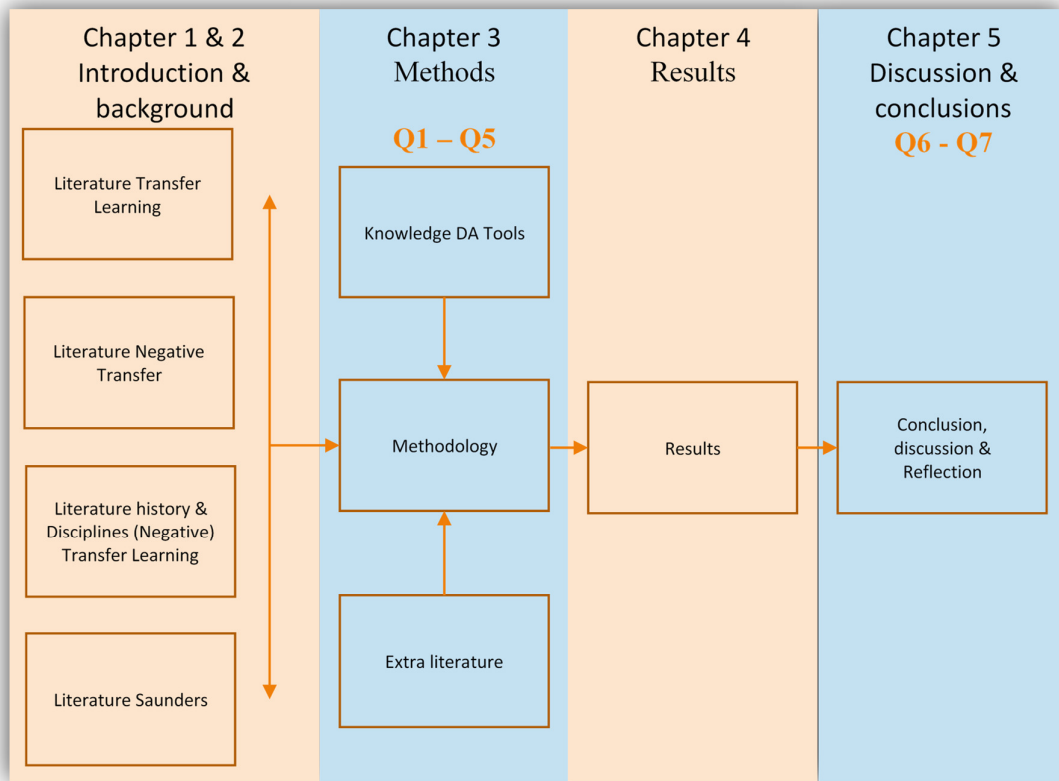


Figure 1 Structure of the research

Chapter 2 describes the research approach for the theoretical framework in which previous research of TL was compared to answer Q1 to Q5. Chapter 3 contains the substantiation of the empirical research that will give the knowledge to formulate the hypotheses for Q6 to Q7. In Chapter 4 are the results of the experiment and Chapter 5 contains the conclusion, discussion and reflection of the research.

2. Theoretical framework

2.1. Research approach

A research should be started with the search for scientific papers and extracting the information (Saunders, Lewis, & Thornhill, 2016). To do this the following steps were taken:

1. Start with four basic papers provided by the thesis supervisor Dr. Bromuri.
2. Literature search in Google Scholar and the Open University Library for peer-reviewed papers, plus additional papers from the thesis supervisor.
3. Second iteration of search in Scholar and Library with the focus on effective TL and NT.
4. Reading and comparing literature and writing down the results.

In the search for useful search terms, the important articles about TL and NT was read. First the articles of S. J. Pan and Yang (2010), Ge, Gao, Ngo, Li, and Zhang (2014), Weiss et al. (2016), Gui, Xu, Lu, Du, and Zhou (2017) were evaluated. Then the valuable objects are and presented in a relevance tree. The objects are included in the search terms.

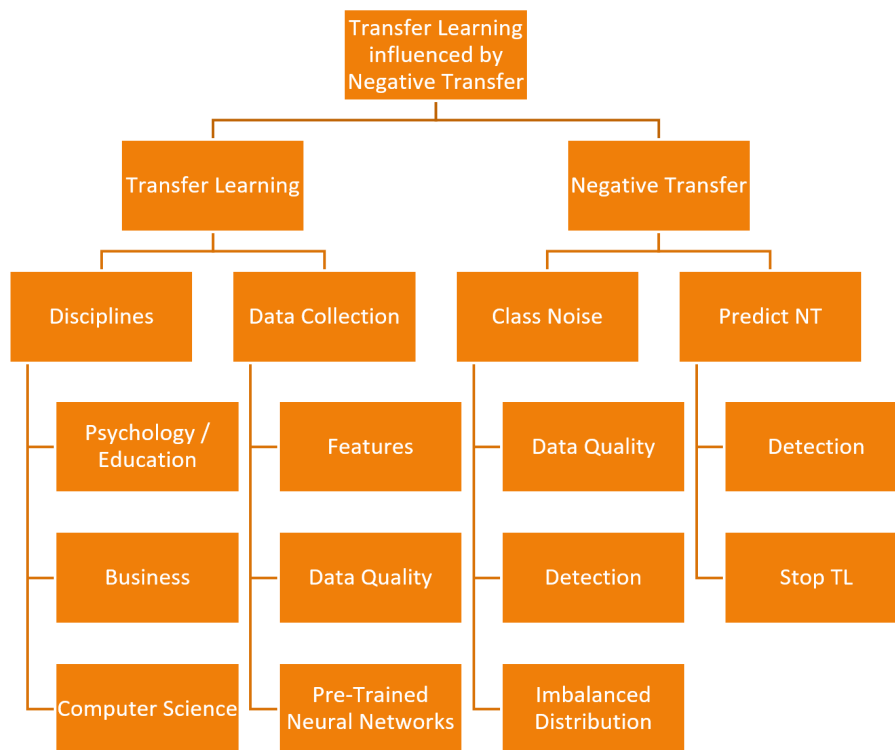


Figure 2 Relevance Tree

To find the essential keywords, a Python Word Cloud Creator (Muller, 2017) was used, based on the number of word counts in the papers. A high number of occurrences suggested an essential word for the subject (Figure 3). Essential words were added to the search terms.

2.2.1. High impact papers

In the table 2 below, are the *high relevant* papers found in the search.

Author, Year, Publisher	Citations	Description and why relevant?
S. J. Pan and Yang (2010), IEEE	3258	Pan et al. can be considered as the most important work on TL. They describe three questions: 1. What to transfer? Which part of knowledge should be transferred? 2. How to transfer? They distinguish three different settings: Inductive TL, Transductive TL and Unsupervised TL. 3. When to transfer? Sometimes knowledge should not be transferred when for instance source and target are not related to another because it could lead to an NT. This paper is interesting because it is a critical survey, with many citations.
Rosenstein, Marx, Kaelbling, and Dietterich (2005), MIT & Oregon State University	196	Rosenstein et al. describe a detection and avoidance method for NT by using very little data from the target task. They describe the challenge for TL is to learn what knowledge is necessary to transfer and how, when to and when not transfer. TL often works well but can also decrease the performance if the source is 'too different'. Rosenheim et al. developed a model, based on naïve Bayes to avoid NT and works well. This is an interesting paper because it provides a way to decide: to transfer or not to transfer.
Torrey and Shavlik (2009), Hershey	166	Torrey et al. write about the concepts of TL and NT. The goal of TL: make ML more efficient. This research is relevant because it explains TL in detail and mentions NT as well.
Weiss et al. (2016), Springer International Publishing	53	This paper is an overview of the TL research field and lists new methods and techniques available. Also, the paper shows there is much research on TL and less on NT and there is much potential in optimizing Transfers. This research is interesting for the thesis because the focus is on current trends in TL.
Ge et al. (2014), Wiley Online Library	22	This research describes potential problems in Multiple Source TL. The two main problems for low performance on TL are NT and Imbalanced Distributions. In the examples mentioned in this paper, the NT occurs when the vector spaces are too different. The Imbalanced distributions have to do with datasets where the classes are not balanced. This paper is relevant because of the examples of NT and the description of ways to prevent it.
Gui et al. (2017), Springer-Verlag Berlin Heidelberg	0	Gui et al. describe ways to detect NT in Transductive TL. It describes the problem of class noise, how it can affect the performance and how it can be detected. Transferred data has a high probability to wrongly be incorrectly classified. They identify high-quality samples to detect class noise in transferred samples, but NT often when the negative reduction sampling is used. This paper is relevant because it presents a method of detecting NT.

Table 2 High impact papers

2.2.2. Medium impact papers

In the table 3 below an overview of the *medium relevant* papers.

Author, Year, Publisher	Citations	Description and why relevant?
Schmidhuber (2015), Elsevier	2046	This paper provides an overview of the workings of Neural Networks (NN's). NN consists of many neurons that create a sequence of value-connections. These neurons are interconnected and attempt to learn behavior patterns, i.e., like driving a car. This is interesting because it relates strongly to how TL works.
Woodworth and Thorndike (1901), American Psychological Association	1420	Woodworth and Thorndike (1901) write about the concept of TL and describe how people can use knowledge learned from previous tasks applied to similar new tasks. The paper is interesting since it assists in understanding the history of TL, but it is not considered core literature for this thesis.
Dai, Yang, Xue, and Yu (2007), Corvalis Oregon	921	This paper presents a TL framework called TrAdaBoost. TrAdaBoost can learn from old data and apply the knowledge (classifications) on new data. The article is interesting for this research because it tries to find a way to improve TL. Since this article is from 2007 and there is not much (peer-reviewed) research on TrAdaBoost is scarce, the importance of this article is classified as medium.
X. Zhu and Wu (2004), Kluwer Academic Publishers	381	This paper researches the impact of class noise and attributes noise. It describes a systematic evaluation of the effect of noise on ML and how to prevent this. Noise can reduce system performance, classification accuracy, extra time building classifier and increase the size of the classifier. So, noise in ML is a problem in many datasets/models. This is highly interesting because noise could be a cause for NT.
Lu et al. (2015), Elsevier	81	Lu et al. assembled a survey on TL in Computational Intelligence. In this article, they distinguish four categories of TL techniques and clusters. A. Neural Networks-based TL B. Bayes-based TL C. Fuzzy TL and D. Computational Intelligence-based TL. This article relevant because of its view on TL and the mentioning of NT.
Kocaguneli, Menzies, and Mendes (2015), Springer	33	Kocaguneli et al. describe a way to predict software defects cross-company and time. They describe the value of old data for organizations as resources for new projects. The combination of effort estimation, TL between time intervals and domains makes this paper relevant.
X. Hu et al. (2016), Elsevier	4	This paper analyses the effectiveness of Multi-Bridge TL and proposed a general cross-domain learning model based on non-negative matrix tri-factorization technology. This model builds multiple latent spaces and learns from multiple bridges knowledge to transfer. The paper is interesting because of the multi-bridge (source) approach. This impact of this article is classified as medium.

Table 3 medium impact papers

2.2.3. Low impact papers

Table 4 below presents an overview of the *low relevant* papers: this does not mean these papers are not essential but rather indicates a lower influence on this research.

Author, Year, Publisher	Citations	Description and why relevant?
Haskell (2000), Academic Press	777	Haskell (2000) addresses the question to teach or not to teach for transfer. It contains material about TL for educators and psychologists and is classified as low relevant for the thesis. Still, it gives insides about the concept of NT.
Leberman, McDonald, and Doyle (2006), Grower	178	The book <i>The Transfer of Learning: Participants Perspectives of Adult Education and Training</i> Leberman et al. write that the transfer of learning is the most critical thing for educational learning and NT happen when tasks are to differ too much from the prior knowledge available. The book views TL and NT from a psychological standpoint. It is an essential book for Learn Psychologists but classified as a low-impact source because it is not peer-reviewed.
Perkins and Salomon (2012), American Psychological Association	102	This paper contains a study 'Transfer of Learning'. "Transfer of learning happens when learning in one context or with one set of materials impacts on performance in another context with other related materials" (Perkins & Salomon, 2012). This paper is useful in answering Q1, but not important enough for the whole thesis.
Do and Ng (2006), Stanford University	100	Do and Ng (2006) discuss the use of TL in text classification. The paper presents a TL algorithm in addressing classifications problems. The research is relevant for TL but does not mention NT.
Griffiths, Johnson, and Mitchell (2011), APS	16	The researchers describe that in the human associative learning there are two guiding principles; the predictive principle which uses outcomes from the past to predict and the uncertainty principle which about not much known yet. This study claims the link between NT and the uncertainty principle by delivering evidence in two experiments on animals and humans.
Helms (2015), Open University Press	not officially published	An exciting oration on Big Data but since it is not a peer-reviewed paper it is only used in Chapter 1.

Table 4 Low impact papers

2.2.4. Sources quality

Table 5 below shows an overview of the quality of the used sources. The H-Index is the sum of the journal's number of articles (h) that received at least the same number of citations. Scimago Journal & Country Rank calculate this amount.

Journal title	H Index	Used articles
Neurocomputing	94	3
arXiv preprint arXiv	-	3
Psychological review (APA)	178	2
Knowledge-Based Systems (Elsevier)	74	2
Advances in neural information processing systems	-	2
Artificial Intelligence (Elsevier)	123	1
Psychological science	207	1
IEEE Transactions on knowledge and data engineering	130	1
Machine learning (Kluwer)	124	1
Pattern Recognition Letters (Elsevier)	122	1
Neural networks (Elsevier)	117	1
Educational Psychologist	99	1
International Journal of Information Management	77	1
Journal of biomedical informatics (Elsevier)	71	1
Journal of Network and Computer Applications	59	1
Artificial Intelligence Review (Kluwer)	54	1
Empirical Software Engineering (Kluwer)	50	1
Knowledge and Information (Springer)	47	1
EURASIP Journal of Wireless Communications and Networking (Springer)	36	1
International Journal of Machine Learning and Cybernetics	23	1
The ASA Data Science Journal	14	1
Journal of Big Data (Springer)	5	1
NIPS 2005 Workshop	-	1
Proceedings of the 24th international conference on Machine learning	-	1
Cognition and instruction: Academic Press	-	1
Advances in Neural Information Processing Systems (2006)	-	1
25th International Conference on World Wide Web	-	1

Table 5 H-index and articles used

2.3. Results and conclusions

Based on the analyzed literature from the previous paragraph, sub-questions Q1-Q5 were answered.

2.3.1. Q1. What is Transfer Learning?

The principles of TL are found throughout the history of humanity. In the stone ages, humans had to learn to transfer knowledge to survive, for instance in finding non-poisonous berries, creating a fire or when building a shelter. This knowledge can stem from earlier or other people's experiences (Deacon & Deacon, 1999). TL in Psychology describes the ability to use previously learned solutions for problems in other similar problems.

TL tries to transfer knowledge. One of the first people to research this were the psychologists Woodworth and Thorndike (1901). They described that how more elements are similar, the better they transfer. This research was a starting point for many other researchers throughout various disciplines. Woodworth and Thorndike (1901) noted that; The definition of TL from a psychological point of view is: “the study of dependency of human conduct learning or performance on prior experience.”. In learning psychology, the transfer performance is the reference to the effectiveness of learning. If people want to use newly learned strategies in different situations, the task must have similar features as the original situation (Leberman et al., 2006).

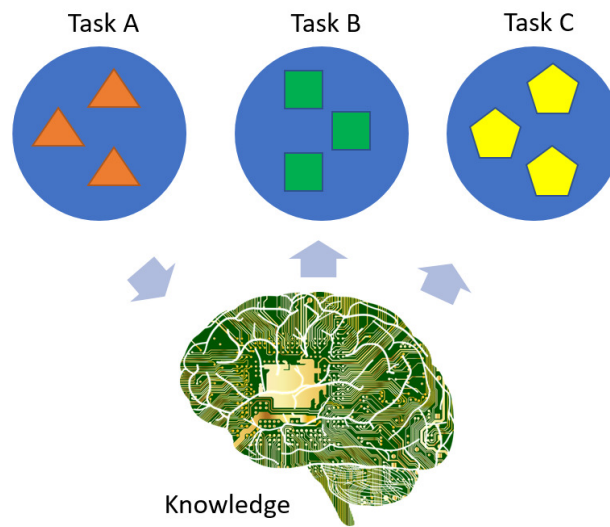


Figure 4 Transfer Learning in psychology

Cognitive skills are required to estimate if a new task is similar enough to make a transfer viable. The skill herein is the ability to discriminate. “Transfer of learning happens when learning in one context or with one set of materials impacts on performance in another context with other related materials” (Perkins & Salomon, 2012).

A task is defined as a problem to solve (Weiss et al., 2016), for instance learning to count. In Figure 4 is task A executed (counting), the experiences (strategies) is saved to knowledge which could be used for counting the other shapes task B & C. In ML this works differently and is answered in Q4.

2.3.2. Q2. What is a Negative Transfer?

NT occurs when previous learning or experience inhibits or interferes with learning or performance in a new context (Leberman et al., 2006). The transfer is positive when experience leads to apprehending a valid or useful similarity relation and improves the performance of learning in some context (Perkins & Salomon, 1992). In general, NT occurs when experience leads to apprehending or applying an invalid similarity relation (Haskell, 2000).

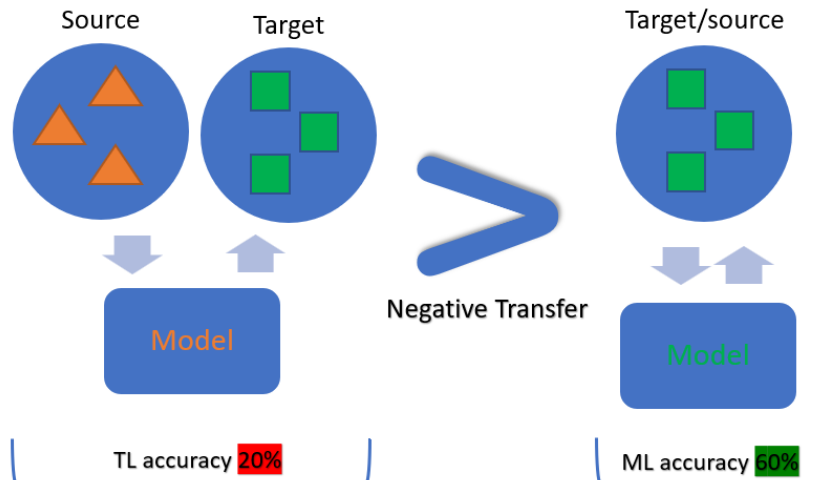


Figure 5 – An example of Negative Transfer happening: learning from the start is works better as TL.

NT can also be the consequence of either insufficient knowledge or prior learning interfering with the current learning task. An NT is when TL is less effective then start learning from the beginning (Griffiths et al., 2011) (Figure 5). Potential cause for this problem could be task similarity and is researched further on in this thesis with a focus on computer science.

The opposite of an NT is a positive transfer (PT) (Figure 6), this happens when TL more effective as learning from the beginning (Perkins & Salomon, 1992).

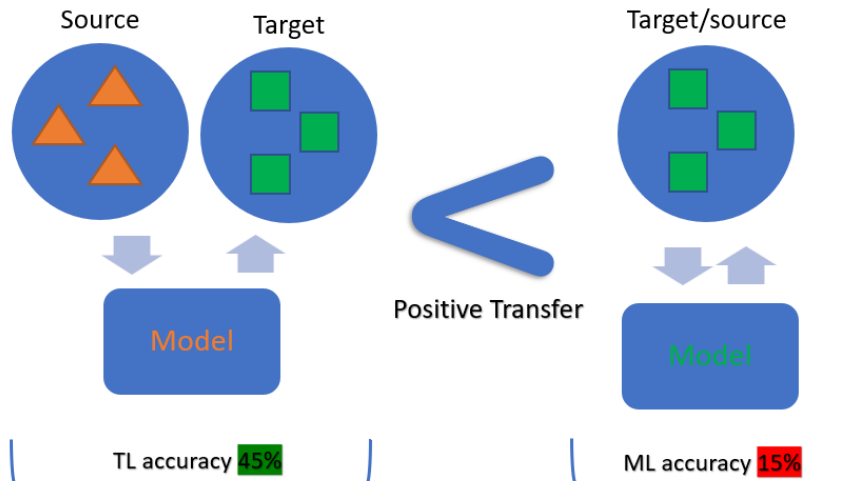


Figure 6 – An example of Positive Transfer happening: using earlier obtained knowledge is 30% higher.

2.3.3. Q3. Which disciplines make use of Transfer Learning?

Many disciplines are researching TL (S. J. Pan & Yang, 2010; Weiss et al., 2016). The Open University Library is used as reference to answer this question, and as source, for “Transfer Learning,” and it classifies the most used disciplines.

TRANSFER LEARNING DISCIPLINES AND RESEARCH

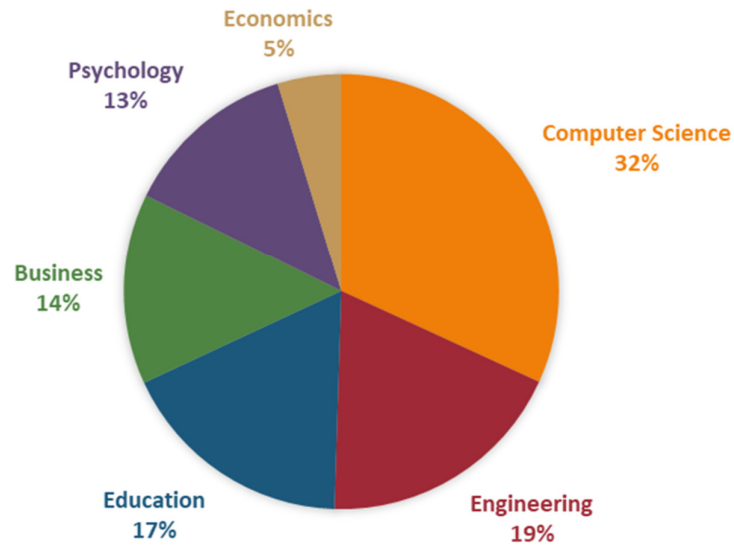


Figure 7 – TL disciplines and research

The most substantial part of TL-research is for Computer Science & Engineering this can be clarified because of the great progression computer algorithms can have from TL.

Education & Psychology use TL to improve human learning. TL is the core business of education, and psychology tries to find out how and how to customize this.

Business and economic researchers are in between, sometimes they use TL to improve their businesses with technology but also to understand their customers/employees and how they learn.

2.3.4. Q4. What does TL mean in computer science?

Traditional ML assumes that training, testing data, and feature space (i.e. the features of a dataset) are in the same distribution (i.e. the dataset), however, not in TL. Situation where is often chosen for a TL approach are: when the source do not have enough data to learn from, the quality of the data is not good enough, or it takes too much time to train the model (S. J. Pan & Yang, 2010).

Figure 8 shows traditional ML. The model needs learn to recognize and count triangles from dataset-A; the performance will probable be good because it uses the same train and test feature space. However, model 1 could not be used in dataset-B because it is trained to count triangles and not squares. The different feature space causes the performance to be poor. Then TL is needed (Rosenstein et al., 2005).

Figure 9 shows TL. The model is trained on a source (dataset-A) and transfers the knowledge to the target (dataset-B). In most situations the results again will be poor because the model only is trained to recognize triangles. To solve this dissimilarity problem a dataset with triangles and squares (more data) could be used or a TL model that combines the datasets (Rosenstein et al., 2005), see Q5. TL looks simple, but in practice, can be hard to transfer the knowledge from one dataset to another dataset (Yu & Deng, 2014).

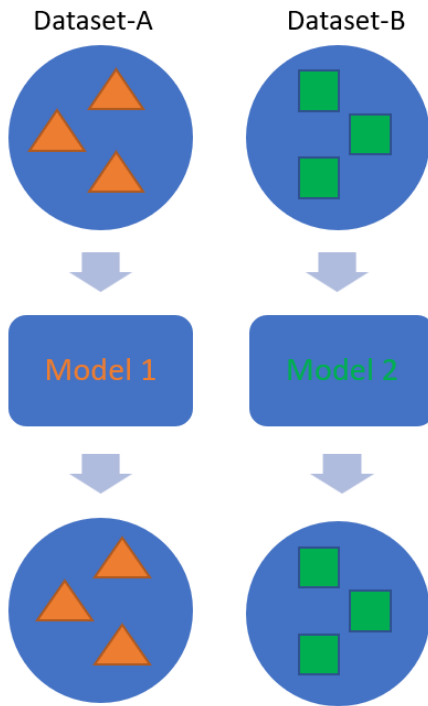


Figure 8 - Traditional ML

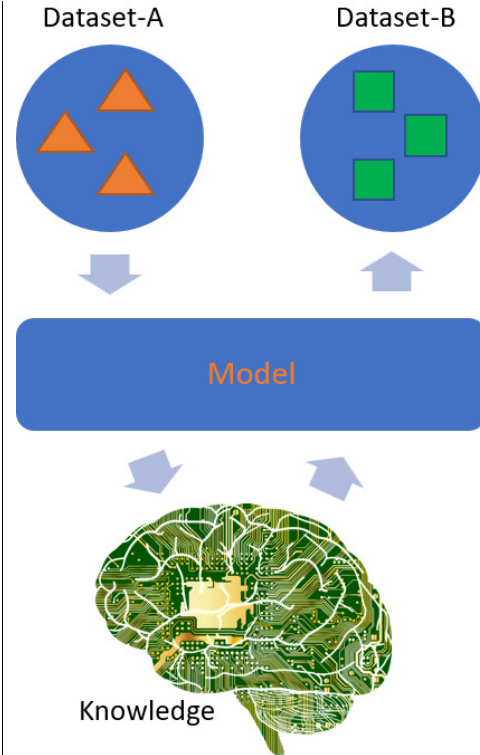


Figure 9 - Transfer learning

In real-world situations, the premise that the feature space is the same will not hold (S. J. Pan & Yang, 2010). TL attempts to overcome the differences and learn the model to reapply the knowledge on different feature spaces. If the feature space of Dataset-A is too different from Dataset-B, this could lead to a NT (Gui et al., 2017).

In short, TL is a way to improve ML and exploit knowledge from a source to a target (Lu et al., 2015). One of the most important papers for TL is S. J. Pan and Yang (2010). This paper provides a definition for TL that will be upheld in this thesis. Being: “Given a source domain and learning task, a target domain, and learning task, TL aims to help improve the learning of the target predictive function using the knowledge using the knowledge given by the source domain and learning task”(S. J. Pan & Yang, 2010).

2.3.5. Q5. What approaches are often used in TL?

S. J. Pan and Yang (2010), create an overview of different approaches for TL and distinguish approaches based on available labels in data and task (Figure 10). When there is labeled data in target domain is TL method is called inductive TL. Within inductive TL is separated self-taught learning where the source labels are unavailable (Raina, Battle, Lee, Packer, & Ng, 2007) and Multi-Task learning or Multiple Source Transfer Learning, where is also available source labels. This is when knowledge can be transferred from multiple sources, this is referred as Multiple Source Transfer Learning (MSTL) (Ge et al., 2014). MSTL has the advantage that there could be learned from multiple sources (Ge et al., 2014) and could improves the dissimilarity problem mentioned in Q4.

Transductive TL is the second TL approach is where the labeled data is only available on the source domain and separate domain adaptation and sample selection bias / co-variate shift (S. J. Pan & Yang, 2010).

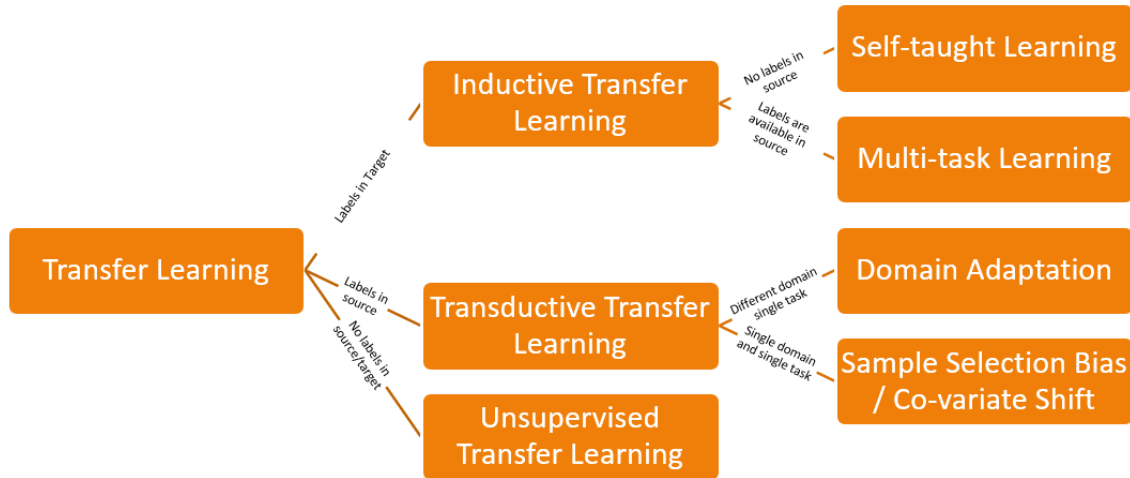


Figure 10 TL approaches Pan & Yang, 2010

Unsupervised TL is the last approach written by S. J. Pan and Yang (2010) about finding clusters, dimension reduction or density estimation within the combined data and can be used when there is no labeled in source or target domain (S. J. Pan & Yang, 2010).

TL Settings	Related Areas	Source Domain Label	Target Domain Labels	Tasks
Inductive TL	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive TL	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised TL		Unavailable	Unavailable	Clustering, Dimensionality reduction

Table 6 TL tasks Pan & Yang, 2010

Beside the approaches the following techniques are also important for TL (S. J. Pan & Yang, 2010):

- Instance-transfer: used for transferring knowledge of instances. It will re-weight labeled data from the source to use in the target domain.
- Feature-representation-transfer: Tries to find strategies to present the features in a new situation. For instance, the date in the source is noted as 01-01-2020 and in the target as 01/01/2020.
- Parameter-transfer: Tries to discover which parameters of the source and target are shared and could lead to a PT.
- Relational-knowledge-transfer: Creates a map of the relational knowledge between source and target.

Another often-used TL-approach are pre-trained models. To build a high performing classifier, is much data and time needed and this is not always available. Pre-trained models could help in this situations (Krizhevsky et al., 2012). Earlier in the thesis is ImageNet mentioned, a database with 1.2 million of labelled pictures. The knowledge about these objects is already pre-trained and could be reused to classify other pictures (Krizhevsky et al., 2012). AlexNet is the neural network connected to ImageNet.

Developers, can connect to the API (application program interface) and use the system to classify own images. Neural networks work with neurons and build cross-linked connections like synapses in the human brain. (Schmidhuber, 2015).

Class noise is not really an TL-approach but relevant for the success of a TL: Several papers showcase that TL is effective when there is not too much class noise during learning iterations, otherwise this could lead to NT (Gui et al., 2017). Real-world data suffers from corruptions (noise) due to data entry failures and poor implemented data acquisitions which can reduce the algorithm performance in classification accuracy, and the time necessary to build the classifier and size of the classifier (creating extra pre-process steps etc.). If class noise could be detected the accuracy of a classifier is likely to improve (X. Zhu & Wu, 2004). Gui et al. (2017) developed a method to predict when the occurrence of NT by sampling the data and estimate the class noise rate of transferred data (Gui et al., 2017).

3. Methodology

Chapter 3 the implementation of the research will be described, Chapter 4 the results and Chapter 5 presents the conclusions and contains a discussion, recommendations, and reflection.

3.1. Conceptual design: select the research method(s)

The quantitative research is about finding answers the sub-questions Q6 and Q7. These answers will be researched by performing a data-analytic experiment, based on CRISP-DM. The CRISP-DM provides a life cycle for data-analytic-projects (Chapman et al., 2000) and will be used to measure the effects of TL, or in short: if TL leads to PT or NT. For this experiment, is worked with mainly datasets with captured reviews from Amazon Review Dataset (He & McAuley, 2016) and Yelp (Asghar, 2016).

The analyses allow the acceptance or rejection of the hypotheses (Saunders et al., 2016) when the probability exceeds the recall/precision of the ML-values. The hypotheses are based on the literature found in Chapter 2 and will be tested with review data.

Hypothesis 0: Semantically related topics have a similar probability distribution, and will have in TL no impact on PT or NT.

Related topics are defined as: transferring reviews of products that are similar to each other, for instance: Amazon categories 'beauty' and 'health and personal care' containing products for the body and is more closely related to each other as automotive or restaurant reviews.

Hypothesis 1: Semantically related topics have a similar probability distribution. Therefore, chances for TL to lead a PT are likely.

Three models are tested; a ML-model which is used as baseline and two TL approaches: Transductive-TL (Domain Adaptation) and Inductive-TL (MSTL).

3.2. Technical design: Elaboration of the method

The CRISP-DM process has six phases that describe the data analytical process (Figure 11). The arrows explain the sequence a that needs to be followed. Sometimes new steps refer to previous steps because of possible new insights (Provost & Fawcett, 2013). The experiment in Chapter 4 will follow the structure of CRIPS-DM (Chapman et al., 2000).

3.2.1. Focus

It is vital to focus on project objectives from a business perspective (Chapman et al., 2000; Provost & Fawcett, 2013) so the first phase is to create business understanding discussed in the next Chapter. The second phase is "Data understanding" (how is set up, data quality problems, first data insights). Third comes the data preparation phase to improve

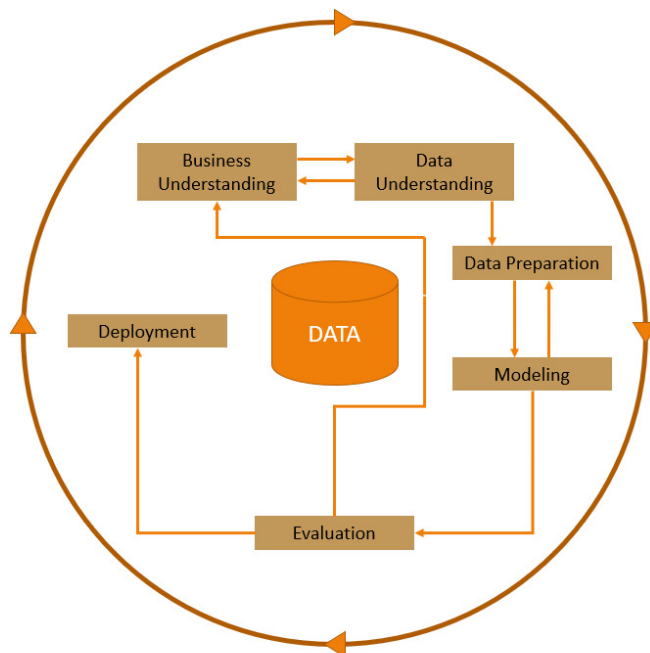


Figure 11 – CRISP-DM process

data quality and perform cleansing. Then the next chapter is about modelling phase, designing the model and, evaluation phase will present the results, and the last is the deployment phase which is meant of implementation which is out of scope.

3.2.2. Tools

Most preparation work and formatting is done in Notepad++, Python, and Ubuntu. For the analysis RapidMiner is used with several extensions. RapidMiner is a convenient tool for doing complex data-analytic work without the need of having large programming knowledge and works with visual workflows.

When RapidMiner is used to classify it produces a confusion matrix which can be used to compare the quality of the model (Figure 12).

	True Negative	True Positive
Predicted Negative	True Negative (TN) Prediction right	False Negative (FN) Prediction wrong
Predicted Positive	False Positive (FP) Prediction wrong	True Positive (TP) Prediction right

Figure 12 Confusion Matrix

3.2.3. Measuring

Similarity measuring in reviews is difficult because the text in the reviews is short which can lead to inaccurate similarities (Lin, Jiang, & Lee, 2014). Therefore, the following metrics are being and compared in the evaluation section:

Value	Description / calculation
Accuracy	Measures the 'effectiveness' of the model and should be treated with caution because the high accuracy in unbalanced classes is often caused majority class. <i>Formula: correct predictions / total predictions * 100, percentage of the classifier quality</i>
Error	Classification error. Opposite of the accuracy and reports the total errors and necessary to calculate the confidence interval. <i>Formula: incorrect predictions / total predictions * 100, the percentage of mistakes made by the classifier</i>
Recall	Measure of completeness of the classifier. A low recall is an indication for probably many FN. <i>Formula: TP / (TP+FN)</i>
Precision	The number of the classifiers rightness or exactness. A low precision is an indication for probably many of FP's. <i>Formula: TP / (TP+FP)</i>
Logistic Loss	The logistic loss of a classifier. The lower the number the less loss a model creates the model. <i>Formula: $\ln(1+\exp(-[\text{conf}(\text{CC})]))$ (where 'conf(CC)' is the confidence of the correct class) (Rapidminer-Website, 2018)</i>
F1 Score	Contains the balance between precision and recall. The higher the number the better the precision and recall are. <i>Formula: $2*((\text{precision}*\text{recall})/(\text{precision}+\text{recall}))$</i>
Confidence Interval	For dataset validation the confidence interval is added. This is presented in a range with an upper and under limit of performance and represent the probability that model will fit in the range. <i>Formula: error +/- const * sqrt((error * (1 - error)) / n)</i>

Table 7 Dataset measuring metrics

3.3. Reflection validity, reliability and ethical aspects

Validity is measuring what you want to measure (Saunders et al., 2016). For this paper are only peer reviewed papers used, classified on relevance and ranked by the publishers and this increases the validity. For the qualitative part, an experiment is being done with real business data and the results being compared the literature which leads to a high validity.

Reliability of the research is about reproducibility. All data is open downloadable, and the workflows are printed in this thesis and could be reproduced.

From an ethical point of view, it could be that review data differs from other texts. These reviews contain no personal information. Sometimes products or company are named so it can be used without any ethical problems.

Another point is that maybe not everybody writes reviews. It is possible that only unsatisfied- or people with other-specific motivations write reviews (fake reviews), this possibly affects the dataset. This will be further discussed in Chapter 5.

4. Results

In this Chapter, the results are presented of the experiment used to accept or reject the hypotheses.

4.1. CRISP-DM: Business Understanding

This section is about the business understanding of the research by the CRISP-DM methodology like discussed in paragraph 3.2.

4.1.1. Determine Business Objective

The business objectives are:

- (1) Find out on what way TL works well.
- (2) Find out how the NT could be prevented

Analyzing the datasets beyond the business objectives or to create an optimized classifier model for a single dataset is not an objective for this thesis.

Three models are developed to predict the star rating and, test the semantically related topics and measure when NT occurs.

4.1.2. Asses Situation

Reviews are an important influencer for customers to buy products and services (Chatterjee, 2001), but not the reason why reviews are used in this thesis. To use inductive and transductive TL labeled data is needed, and this is provided in reviews from Amazon and Yelp.

For gathering information from unstructured data, text mining is applied. Text has a linguistic structure, and from here difficulties arise (Sebastiani, 2002). People make mistakes, use synonyms, abbreviations or place text in a different context, this creates challenges for ML (Jurafsky, 2000). Text mining algorithms like TF-IDF, which counts result in a vector with thousands of attributes lead to massive memory consumption. Therefore, 'Latent Semantic Indexing' (LSI) will be used, this is a topic modeling technique that combines topics where the reviews are about and reduce the resources use (Landauer, Foltz, & Laham, 1998). LSI, can be used in several business cases like information discovery, automated text summarization or spam filtering. There is also a probabilistic-LSI variant, based on the likelihood principle and reduces the word perplexity and shows promising results but this is not yet implemented in RapidMiner (Hofmann, 2017).

To compare TL with traditional ML the modeling the data-analyze is split up into three stages. In (traditional) ML-stage a model is created to measure confidence intervals and metrics on a single dataset. For a full-TL (domain adaptation TL), the same values are measured, then the knowledge is transferred from a source dataset to a target dataset. In a combined-TL (MSTL) is knowledge transferred from the source but also 70% of the target data is used.

4.1.3. Datasets

Data is collected from the existing datasets, "health and personal care" (HPC), "beauty" and "automotive" of Amazon Review Dataset (He & McAuley, 2016) and extracted from the Yelp Review Dataset (Asghar, 2016). From these datasets, a sample of 10.000 records is taken and pre-processed for further analysis.

The use of an existing dataset has the advantage of saving time. The disadvantage is that trust in the datasets for correctness and completeness is necessary. Both datasets used are common peer-reviewed databases (Asghar, 2016; He & McAuley, 2016).

4.2. CRISP-DM: Data Understanding

Data is raw material with strengths and limitations that should be investigated in the data understanding phase (Provost & Fawcett, 2013).

4.2.1. Describe and collect initial data

Amazon is the world largest e-commerce retailer where consumers can buy products and leave a review about products. The Amazon Review Dataset contains 24 product categories (sub-datasets) ranging from books to instant videos. Yelp is a hospitality recommendation and review site where customers can give stars for a restaurant based on their experiences.

Both datasets have a 5-star scale system, where “1” is the poorest and “5” the highest rating.



The Amazon datasets (JSON format) are converted into CSV files which are compatible with RapidMiner.

The Yelp sample is extracted from the “review” table of the SQL database of the Yelp dataset and saved as CSV file (SQL Syntax: select * from reviews order by rand 10000).

After collecting the data, the structure and features are examined. The Figure 15/16 below shows the structure of an Amazon and a Yelp example.

```
1 {
2   "reviewerID": "A2SUAM1J3GNN3B",
3   "asin": "0000013714",
4   "reviewerName": "J. McDonald",
5   "helpful": [2, 3],
6   "reviewText": "I bought this for my husband
7   playing these old hymns. The music is at
8   published for singing from more than playin
9   "overall": 5.0,
10  "summary": "Heavenly Highway Hymns",
11  "unixReviewTime": 1252800000,
12  "reviewTime": "09 13, 2009"
13 }
```

Figure 15 – Amazon example review

```
1 "reviewText";"id";"business_id";"user_id";"o
2 erall";"date";"useful";"funny";"cool"
3 "i love this place. the beer is good, the
4 growlers are cheap, and the staff is
5 friendly. and in a city where beer is as
6 hard to come by as it is in Pittsburgh,
7 it's comforting that local breweries like
8 this
9 exist."; "pt1oOJApGd_kidiuzTQpHg"; "o_g2Q64F
10 c_Q4070EhbQ"; "kxj0GGMLKRILpQvb7TpiJw"; 5.0; 1
11 9/10 12:00 AM; 0.0; 0.0; 0.0
```

Figure 16 – Yelp example 5-star review

Amazon datasets (He & McAuley, 2016)		Yelp dataset (Asghar, 2016)	
Feature	Use	Feature	Use
reviewerID	ID of the reviewer (textual)	text	original text, text of the review (textual)
Asin	ID of the product (textual)	id	Id of the review (textual)
reviewerName	Reviewer name (textual)	business_id	ID of the restaurant/hotel (textual)
X1/X2	original helpful, helpfulness rating of the review (numerical/categorical)	user_id	ID of the reviewer (textual)
reviewText	text of the review (textual)	stars	original stars, 5-star rating of the product (numerical/categorical)
Overall	5-star rating of the product (numerical/ categorical)	date	time of the review (datetime)
Summary	summary of the review (textual)	useful	helpfulness rating of the review (numerical/categorical)
unixReviewTime	Unix time of the review (datetime)	funny	funniness rating of the review (numerical/categorical)
reviewTime	Raw time of the review (datetime)	cool	coolness rating of the review (numerical/categorical)

Table 8 Comparison of the dataset features

4.2.2. Explore data

To increase the data understanding, some statistics are presented here regarding the (sub)-datasets. For all word counting, Ubuntu Word Count is used and counting of character is done in RapidMiner.

	Original datasets				10k Sample datasets		
	Original Number reviews	Original Number of Words	Original Size of the dictionary (Bytes)	Original Words per review	Largest review in char. (sample)	Average review length in char. (sample)	Deviation reviews length in char. (sample)
Health & P.	346.355	34.363.995	214.308.746	99,2	16093	510.146	596.108
Beauty	198.502	18.392.223	114.344.364	99,7	8257	459.054	470.176
Automotive	20.473	2.329.923	11.435.635	113,8	11371	457.426	548.231
Yelp	100.000	11.502.464	63.061.649	115,0	4993	618.241	575.838

Table 9 General statistics

Besides the standard statics are also the number of unique words and deviation relevant. It can show the similarity of the word distribution of the datasets, which probably lead to better transfers (see hypothesis).

	Minimal number of unique words	Maximal number of unique words	Average number of unique words	Deviation number of unique words
Health & P.	0	17,718	6,045	2,199
Beauty	1	18,543	5,961	2,132
Automotive	1	20,712	5,894	2,169
Yelp	1,404	18,780	6,742	2,413

Table 10 Number of unique words & deviation

The number of unique words is counted using the “generate aggregation” operator in RapidMiner. Some notable results on first sighting: Amazon Health & Personal Care (HPC) contains three reviews without text, all Amazon datasets contain one-character or one-word reviews and the average number of unique words in Yelp is the longest.

The Amazon datasets have ~10% of long reviews over 1000 characters, in Yelp more than 17% of the reviews are lengthy reviews. The average length of the reviews is longer on Yelp than in Amazon reviews, but the largest reviews are found in the Amazon dataset. The HPC reviews, are on average slightly longer than Beauty and Automotive reviews.

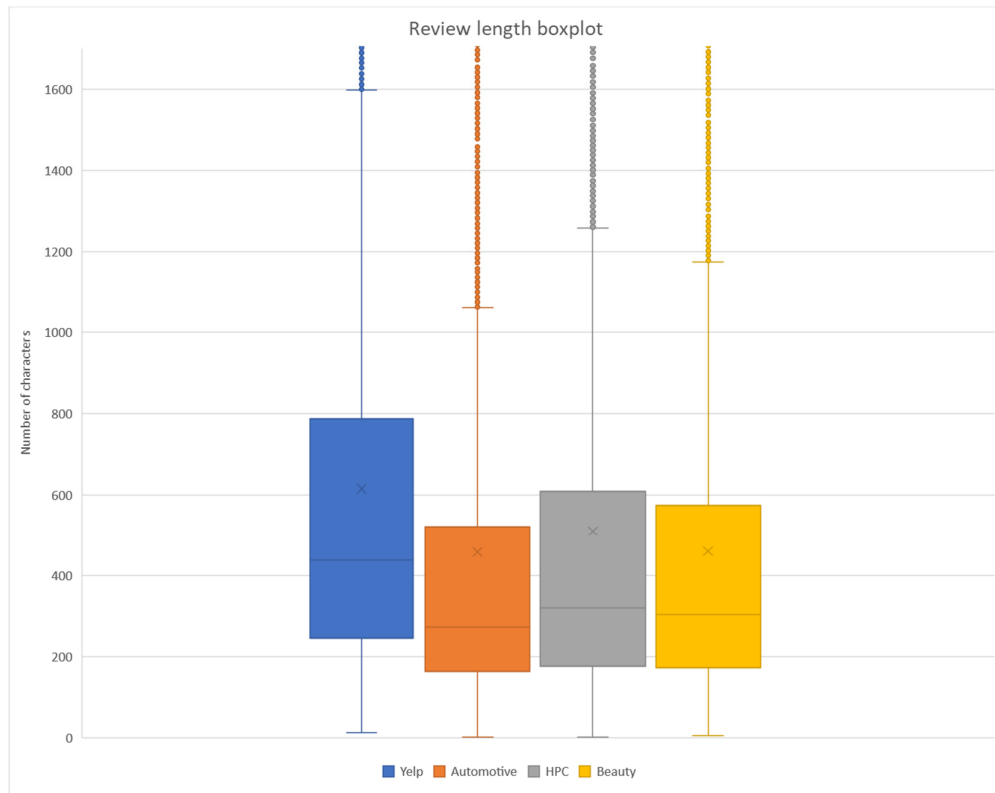


Figure 17 – Deviation of Text length per review and per dataset

Based on the deviation beauty and HPC are similar in length of text and Yelp and Automotive appear less closely related to each other. A relation between lengthy reviews and negative/positive sentiment reviews cannot be found.

4.2.3. WordNet: Most commonly used words

These are the top10 most commonly used words in the datasets, including the number of occurrences in the database. Stop words and words of under two characters are filtered out, words are all converted to lowercase and stemmed:

	Health & Pers.	Beauty	Automotive	Yelp
1	Product (5007)	Hair (8264)	Work (4494)	Good (3808)
2	Work (4136)	Product (7134)	Great (3496)	Food (3729)
3	Take (3196)	Skin (5045)	Product (2923)	Place (3655)
4	Good (3129)	Look (3344)	Good (2873)	Great (3125)
5	Time (2785)	Love (3286)	Time (2441)	Time (3111)
6	Great (2723)	Work (3201)	Look (2185)	Servic (2758)
7	Us (2336)	Great (3120)	Batteri (1965)	Order (2748)
8	Make (1873)	Good (2809)	Make (1911)	Love (1619)
9	Clean (1835)	Us (2795)	Us (1893)	Come (1599)
10	Help (1824)	Color (2767)	Easi (1878)	Nice (1508)

Table 11 Most used words, counting the occurrences of words

All Amazon datasets revolve around several products. This can be noticed by common words like product and work. Yelp is about services, causing words like food, place, time, service. In all datasets, people use words like great and good.

The method above does not show the semantic and synonyms of text. Therefore, WordNet is used (Miller, 1995). The following operators are used to create the table seen in the figure below: WordNet stemmer, find hypernyms, hyponyms, and synonyms. WordNet database v3.1 is used to find the top 10.

First the top10 synonyms (other words with the same meaning), like 'rabbit' or 'bunny':

	Health & Pers.	Beauty	Automotive	Yelp
1	Take (76868)	Make (71784)	Light (64497)	Place (89619)
2	Get (59429)	Dry (60287)	Make (58376)	Good (88440)
3	Make (55277)	Get (51443)	Good (49783)	Get (79357)
4	Good (53154)	Give (49226)	Work (48470)	Go (65752)
5	Give (48390)	Good (47487)	Get (47603)	Make (56148)
6	Work (47043)	Use (43482)	Use (33908)	Give (53452)
7	Clean (33691)	Light (35528)	Clean (33225)	Come (49486)
8	Use (33162)	Feel (32912)	Give (31194)	Take (48692)
9	Cut (29902)	Hair (32903)	Easy (30304)	Servi (41013)
10	Go (24538)	Work (32039)	Take (29537)	Call (38247)

Table 15 Find synonyms based on WordNet

Here are the top10 hypernyms (a word with broad category of other words), like 'primate', belongs to rabbit but also human. So, if RM finds the word 'rabbit' it reports, 'primate'.

hyper:point hyper:section hyper:characteristic hyper:blemish
hyper:marking hyper:section hyper:place_of_business
hyper:occupation hyper:attack hyper:small_indefinite_quantity
hyper:marker hyper:lamp hyper:playing_card
hyper:mistake hyper:spy hyper:dirty hyper:change_surface
hyper:change hyper:mark hyper:meal hyper:eat
hyper:feed hyper:seafood hyper:decapod hyper:Mexican hyper:dish
Example of a processed record in hypernyms

	Health & Pers.	Beauty	Automotive	Yelp
1	Change (66634)	Change (71323)	Change (62975)	Change (76972)
2	Be (59024)	Be (53803)	Be (53974)	Be (71054)
3	Move (28092)	Make (26531)	Move (31540)	Move (36186)
4	Act (27982)	Act (25658)	Make (24997)	Act (31936)
5	Make (26722)	Move (22314)	Act (22769)	Make (29742)
6	Get (25240)	Time_period (18149)	Get (17901)	Experience (25799)
7	Time_period (22499)	Get (18135)	Activity (16880)	Time_period (24118)
8	Experience (20172)	Touch (15185)	Travel (15897)	Travel (20117)
9	Have (17523)	Remove (14821)	Compartment (15241)	Change_state (19496)
10	Activity (17379)	Activity (14810)	Have (14849)	Get (19164)

Table 12 Find hypernyms based on WordNet

Above are the hypernyms that were found. Due to the broad categories, it makes sense that there are fewer differences between the datasets. Words like change, be, move, make and act are in all top 5 lists. More specific are the words after position five, like experience, compartment, and change_state.

hypo:worship hypo:love hypo:get_off hypo:romance hypo:take
hypo:grave hypo:boatyard hypo:level hypo:behalf
hypo:home_away_from_home hypo:academicianship hypo:wing
hypo:niche hypo:perch hypo:postposition hypo:margin
hypo:insert hypo:superordinate hypo:address hypo:distinguish
hypo:fund hypo:garrison hypo:draft_beer
hypo:common_good hypo:kindness hypo:worthiness hypo:basic
hypo:newsroom hypo:alpenstock hypo:crosier
hypo:man hypo:national_capital hypo:draft_beer hypo:milt
hypo:emanate hypo:land hypo:fall hypo:work_out
hypo:aggregate hypo:anesthyl hypo:brewpub hypo:come hypo:breathe

Example of a processed record in hyponyms

A hyponym is opposite from hypernym (a subcategory of a general class), animal has hyponyms fish and primates.

	Health & Pers.	Beauty	Automotive	Yelp
1	Catch (11077)	Take (9409)	Give (7560)	Catch (11606)
2	Take (8990)	Give (9244)	Catch (6888)	Take (11169)
3	Give (7623)	Stinging_hair (8231)	Take (6115)	Sober_up (10402)
4	Decide (6179)	Bristle (8224)	Decide (5290)	Land (10020)
5	Sober_up (6144)	Coat (8224)	Land (5145)	Superordinate (7410)
6	Land (5948)	Ingrown_hair (8224)	Double (5122)	Suffer (7395)
7	Render (5819)	Hollywood (7637)	Render (5114)	Decide (7390)
8	Test (5770)	Catch (7388)	Ambulance (5036)	Render (7266)
9	Misread (5345)	Book (7122)	Sober_up (5036)	Know (6799)
10	Recommence (5266)	Cargo (7116)	Follow (4944)	Academicianship (6597)

Table 13 Find Hyponyms based on WordNet

In all datasets, the most used words are catch, take, give. Often used words sober_up, decide, land and render. More characterizing hyponyms are stinging_hair, academicianship, ingrown_hair, misread, superordinate and recommence.

For instance, the hyponym 'academicianship' belongs to synonyms like position, post, place, situation and hypernyms like job, occupation, business.

4.2.4. Class imbalance

The classes in all datasets are unbalanced as can be seen in the figure under here. In all datasets are the 5-star-ratings are the largest. The Yelp dataset is relatively the most 'balanced' compared to the other datasets, but is still unbalanced.

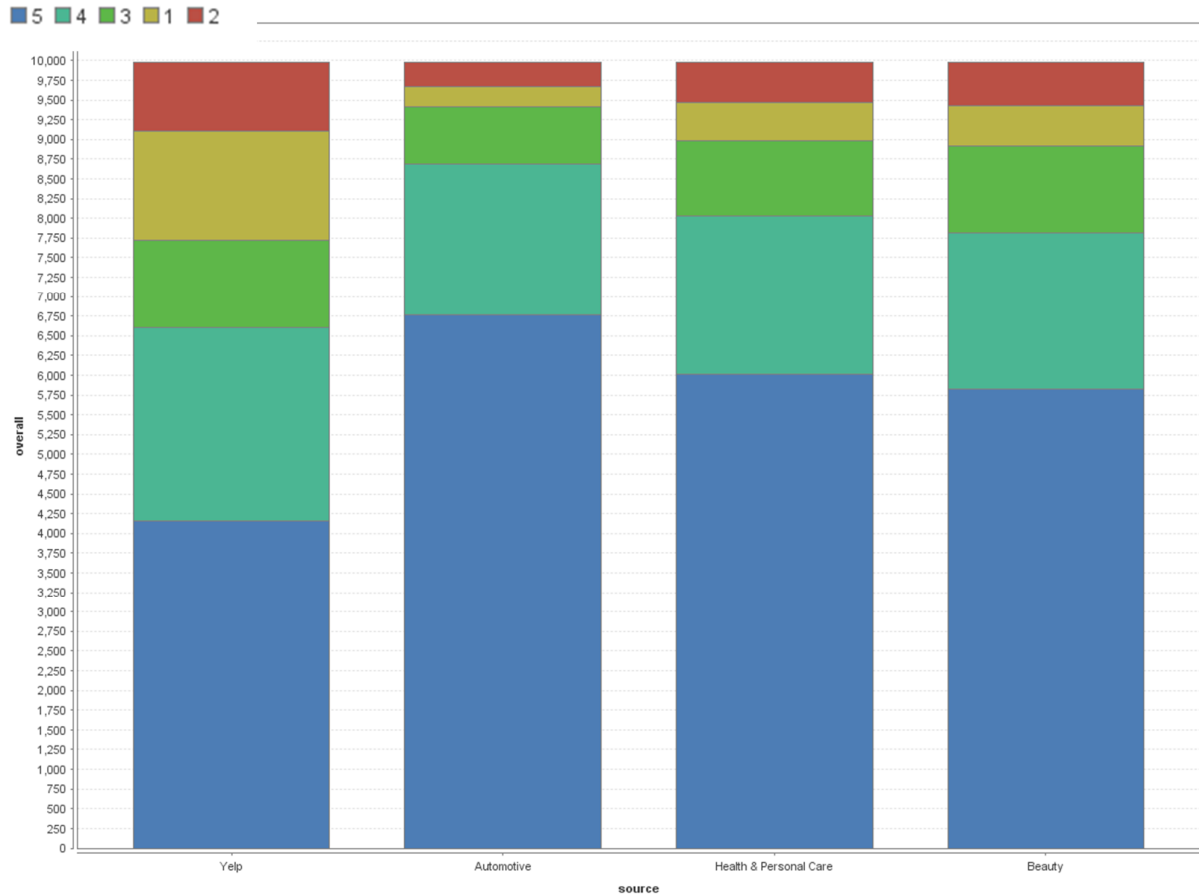


Figure 18 - Class imbalance problem

Removing the lengthy reviews or removing non-English-reviews does not solve the class imbalance problem, so they must be sampled in the model. The automotive dataset is the most unbalanced with 67,5% five-star reviews and 2,6% of one-star reviews.

The high number of five-star ratings are interesting because, if nothing is done, it will influence the classifier of TL later. If customers see already high reviews from other users, they are more likely to give high a ranking to, even if they are not 100% satisfied (Anderson, 1998). In social media marketing this is called a positive eWOM (electronic word-to-mouth), and sellers of products try to influence sites with reviews (N. Hu, Pavlou, & Zhang, 2006).

Some classes are so rare that the prediction of the star rating is unreliable as can be seen in the following table and confusion matrix, also discretizing to only positive/negative classes is not desirable because of the loss of information.

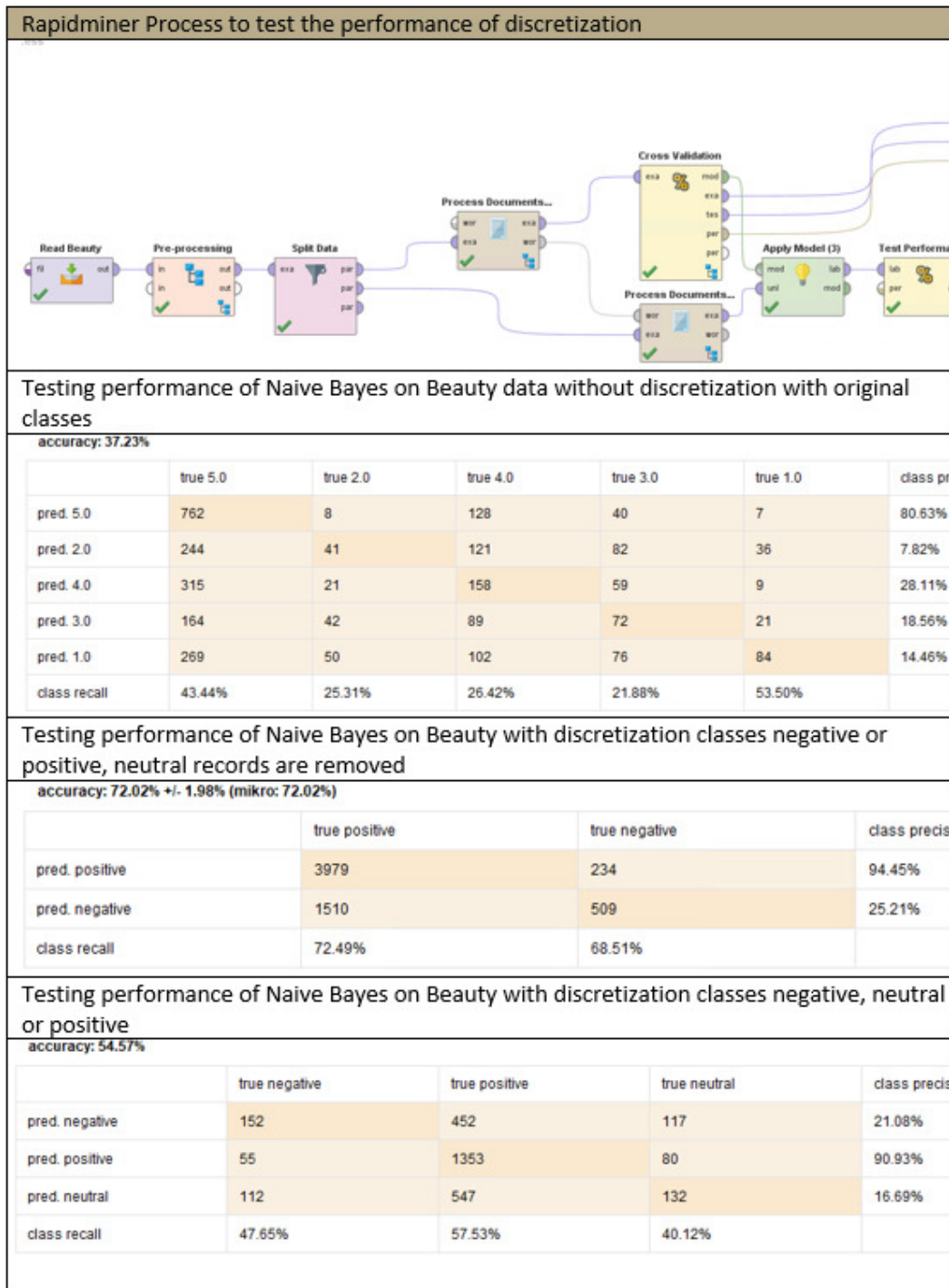


Table 14 Discretization of the classes

If the datasets are getting discretized to three groups: positive, neutral and negative, the dimension of the class imbalanced is clearer. In figure 16, Yelp misses some examples which will be discussed in data preparation phase.

The discretization comes with a small loss of information which is arguably inevitable with these unbalanced datasets.

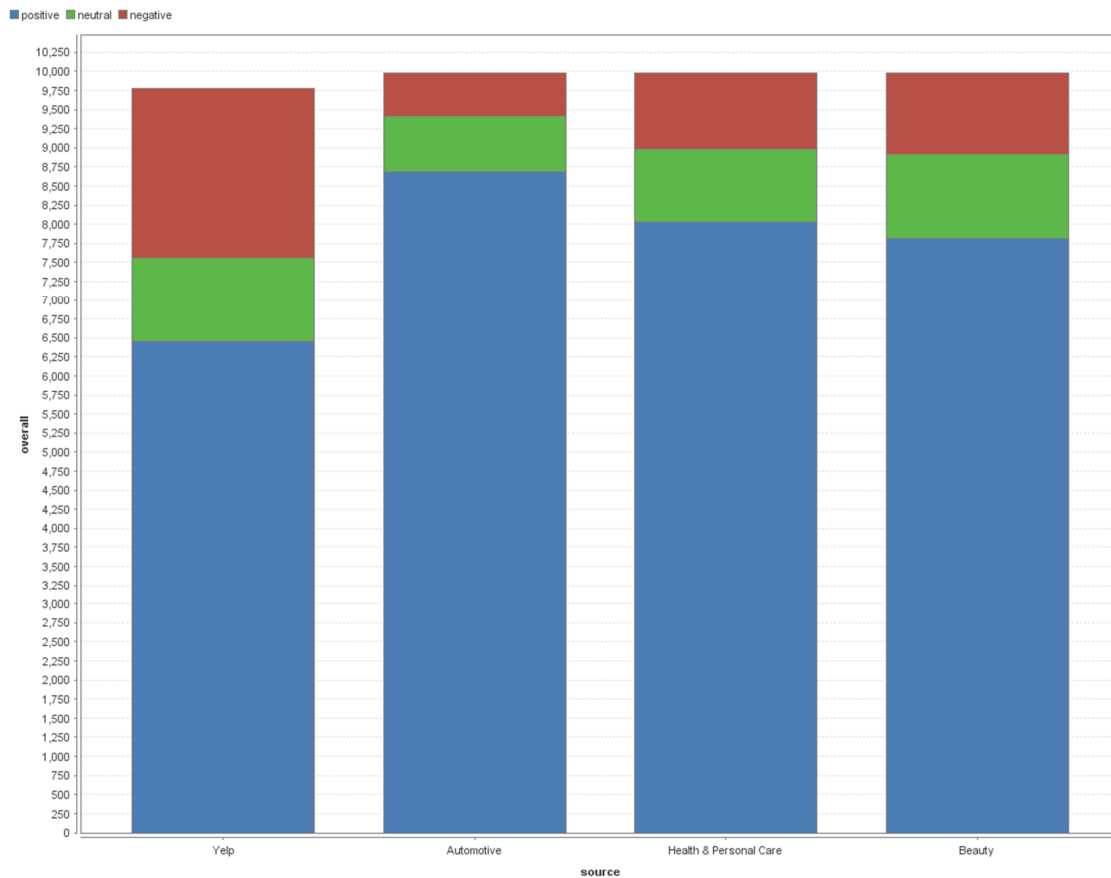


Figure 19 - Class imbalance after discretization

On this way, the datasets are stored and used in the further process in the next CRISP-DM steps, more about this in paragraph 4.3.3.

4.2.5. Missing values

There are missing values in the datasets. Sometimes due to special characters that the RapidMiner parser or CSV-converter sees as an error.

There are a few strategies to work with missing values, remove these records, continue with an average (calculated on the data) or fixed value (Provost & Fawcett, 2013). Due to the large dataset is chosen to remove the records because the few missing records are not expected to be vital. In some cases, the reviews contain no text, for instance just an emoticon like “:-)” or just a character or one single word which leads to fewer quality classifications and can affect TL, so there are filtered out.

All Amazon datasets contain questions marks in the attribute ‘helpful’, because this is not mandatory fields on the website. Yelp also don’t oblige the attributes ‘useful’, ‘funny’ and ‘cool’.

4.2.6. Correlation matrix

In Yelp the matrix shows that the attributes useful (-0.068), funny (-0.046), cool (-0.365) correspond well the label overall. The Amazon datasets have something similar to the 'helpful' rating (range between -0.2 and -0.35). These scores are low because of the non-obligate character and the different use of the datasets not or minor usable.

4.3. CRISP-DM: Data Preparation

Data preparation is used to clean and prepare the datasets.

4.3.1. Clean data

For this thesis is text mining used, and the focus will be on the primary features 'reviewText' (review-text) and 'overall' (star-rating).

The Yelp dataset needed the most cleansing due to the SQL export and RapidMiner import. The data cleansing is done in Notepad++ with regular expressions, for instance, cleaning up line breaks, missing quotes or too many quotes (quote in the text) but there are also other problems with data quality.

The Yelp sample contains reviews in non-English that are not tagged in the originally extracted table or other tables in the Yelp database and are classified by the 'Language Detection API' (Makūnas, 2018).



Figure 20 - Detecting Language in RapidMiner

First, the CSV is read, then a copy of the reviewText attribute is made and named 'message' and encoded to UTF-8. Every review is send to the web service and the added attributes 'isReliable', 'confidence' and 'language' are added. Again, everything is written back to a new CSV file. The confidence gets lower when names of restaurants, products are used, or the text contains multiple typos but always enough to achieve a 'true' at isReliable.

message	isReliable	confidence	language
There%27s+lots+of+potential+here+because+of+...	true	13.440	en
Vendredi+midi%2C+heure+lunch%2C+c%27est+...	true	8.810	fr
The+place+has+been+around+for+a+while+but+i...	true	13.330	en
Tourist+visiting+the+area.+Decided+to+check+thi...	true	11.850	en

Figure 21 - Examples of detected language

1,4% of the Yelp reviews contain other languages as English; German (94), French (40), Spanish (3), Japanese (2) or Tagalog (1) and these reviews will be filtered out before the analyses.

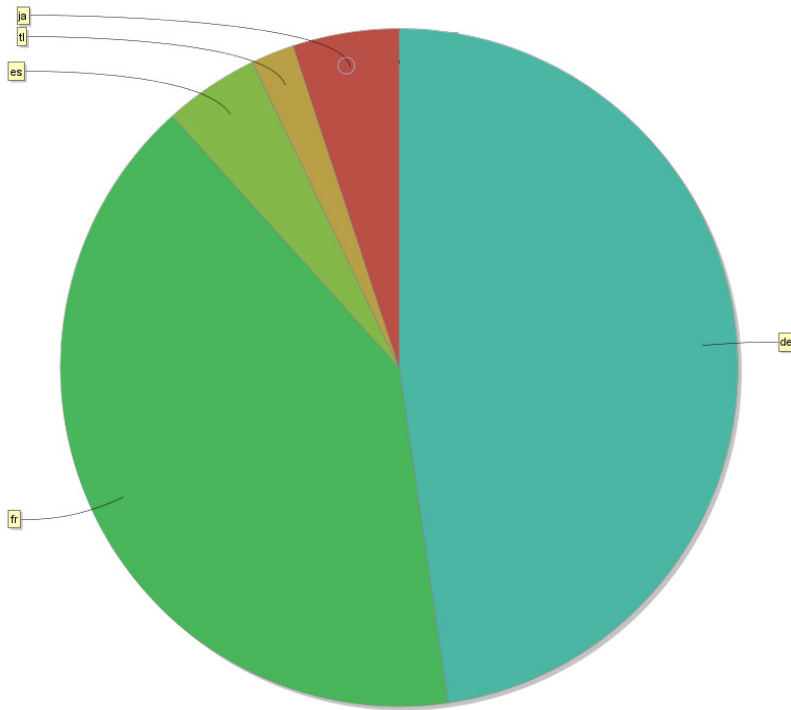


Figure 22 Proportions of other languages as English in Yelp

4.3.2. Construct & integrate data

In order to generalize the main features of the Yelp dataset the attributes “Text” and “Stars” are renamed to “reviewText” and “overall”. The first operator reads the CSV file; after that the relevant attributes are selected and renamed. Finally, the corrected data is stored in a updated CSV file.

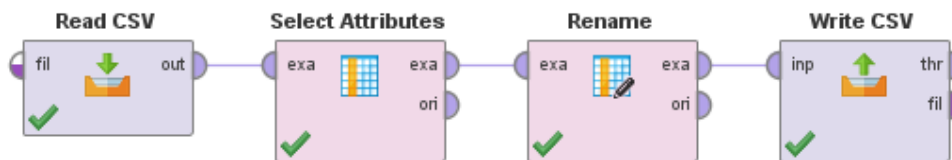


Figure 23 Rename features

The datasets features after renaming them:

Amazon datasets (He & McAuley, 2016)		Yelp dataset (Asghar, 2016)	
Feature	Use	Feature	Use
reviewerID	ID of the reviewer (textual)	reviewText	original text, text of the review (textual)
asin	ID of the product (textual)	id	Id of the review (textual)
reviewerName	Reviewer name (textual)	business_id	ID of the restaurant/hotel (textual)
X1/X2	original helpful, helpfulness rating of the review (numerical/categorical)	user_id	ID of the reviewer (textual)
reviewText	text of the review (textual)	overall	original stars, 5-star rating of the product (numerical/ categorical)
overall	5-star rating of the product (numerical/ categorical)	date	time of the review (datetime)
summary	summary of the review (textual)	useful	helpfulness rating of the review (numerical/categorical)
unixReviewTime	Unix time of the review (datetime)	funny	funniness rating of the review (numerical/categorical)
reviewTime	Raw time of the review (datetime)	cool	coolness rating of the review (numerical/categorical)

Table 15 Overview of the final features

4.3.3. Further cleaning

No duplicate records are found. Outliers are not relevant in this stage because we are using text mining and the text is written by people which is always relevant for the star rating. Outliers could be relevant if people write about non-relevant subjects in the review but this unlikely and manually not detected.

4.4. CRISP-DM: Modelling

This Chapter is about developing the models for the experiment.

4.4.1. Selected techniques

There are three models developed for the experiment with different TL-approaches and all use eight steps to predict the star-rating:

- ML model which is used to gather baseline information, like optimal parameters for datasets.
- Full-TL model (Domain Adaptation TL) so see how a model performs on another dataset, where no content of the target is used. In literature is this called Transductive-TL with domain adaptation (S. J. Pan & Yang, 2010). The labels are only used from the source domain.
- Combined-TL model (MSTL) where, source and target combined to support the predictions. This is called Inductive-TL and more specific MSTL also known as Multi-Task Learning (S. J. Pan & Yang, 2010). Labels and data from source and target are available for learning and applied to improve the predictions.

The most important selected techniques for the models are:

An **elaborate explanation** including operators and parameters can be found in **Appendix 1**.


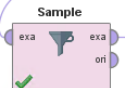
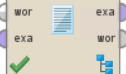
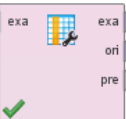
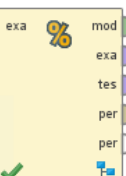
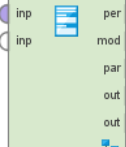
 	<p>Sampling: SMOTE <i>Synthetic Minority Over-sampling</i>, Technique is neighborhood-based to find for the minority class the nearest neighbors and is considered as an upsampling technique that improves inbalanced datasets (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).</p> <p>Downsample, because the datasets are so unbalanced the sample operator use only 30% of the 5-star reviews.</p>																														
	<p>To transform unstructured data into structured data, the operator “Process Documents from Data” is used. This operator is available in RapidMiner after installing the extension “Text Processing’ and is also a subprocess.</p> <p>A simplistic approach in text mining is Bag of Words. This approach treats every document as a collection of individual words (tokens) from the corpus and is useful but ignores grammar and importance of words. A second approach is Term Frequency (TF) this counts how often a word occurs in a document, the more often this appears in the document, the more important it is.</p> <p>For instance when the adjective word “great” is found in more documents then this is probably more important than the word “adapt” which occur only a few times in the corpus. Even more advanced is TF/IDF, based on the multiplication of TF and inverse document frequency (IDF) and indicates how characteristic a term is for the corpus. Words should not too rare but not too familiar either (Engels, 2017).</p> <p>TF-IDF: Term Frequency and Inverse Document Frequency, create a vector based on how often a term occurs divided by the inverse document frequency to increase the weight of characterizing words for every example (Blei, Ng, & Jordan, 2003).</p>																														
	<p>The TF/IDF results in thousands of features. To perform dimensionality reduction the SVD operator (Singular Value Decomposition) is used. This RapidMiner operator uses ‘latent semantic indexing’ which is a method for finding topic models at large corpora of text, and convert the vector to topic models which represent the same topics (Landauer et al., 1998).</p> <p>Optimizing parameter operator showed that 100 dimensions is the optimal parameter.</p>																														
	<p>Cross validation with classifying algorithms kNN, Naïve Bayes (NB) or GLM. These algorithms make the actual prediction of the class en, therefore, it is important to test multiple algorithms.</p> <div><div>ExampleSet (3000 examples, 5 special attributes, 100 regular attributes)</div><div>Filter (3,000 / 3,000 examples)</div><table><tr><th>Row No.</th><th>overall</th><th>prediction(o...</th><th>confidence(...</th><th>confidence(...</th><th>confidence(...</th><th>svd_1</th><th>svd_2</th><th>svd_3</th><th>svd_4</th></tr><tr><td>1</td><td>negative</td><td>neutral</td><td>0</td><td>0</td><td>1</td><td>0.002</td><td>-0.001</td><td>0.001</td><td>-0.002</td></tr><tr><td>2</td><td>positive</td><td>neutral</td><td>0</td><td>0</td><td>1</td><td>0.005</td><td>-0.003</td><td>-0.004</td><td>-0.002</td></tr></table></div>	Row No.	overall	prediction(o...	confidence(...	confidence(...	confidence(...	svd_1	svd_2	svd_3	svd_4	1	negative	neutral	0	0	1	0.002	-0.001	0.001	-0.002	2	positive	neutral	0	0	1	0.005	-0.003	-0.004	-0.002
Row No.	overall	prediction(o...	confidence(...	confidence(...	confidence(...	svd_1	svd_2	svd_3	svd_4																						
1	negative	neutral	0	0	1	0.002	-0.001	0.001	-0.002																						
2	positive	neutral	0	0	1	0.005	-0.003	-0.004	-0.002																						
	<p>Optimizing parameter operator is a subprocess to find the optimal parameter settings, more about operator this in the next paragraph.</p>																														

Table 15 Important selected techniques

4.4.2. Find optimal values & operators

For a good performing model, it is necessary to have optimized operators and parameters. First is started with creating a baseline ML-model and the classifying algorithms are selected, based on the best-performing algorithms coming out of 'auto model' of RapidMiner.

To find the optimal parameters for all source datasets the operator 'Optimize Parameters' is selected. This is a subprocess wherein operators and parameters have tested to find the optimal values and show this results in several iterations. The most important parameters that changed where:

- kNN number of K
- Naïve bayes, laplace_correction
- Cross validation folds
- SVD (LSI) dimensions
- Samples sizes

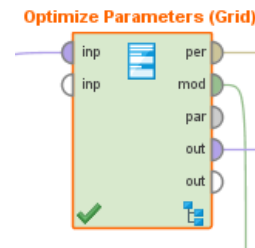


Figure 25 Optimize parameters operator

After kNN optimization is seen that at most datasets the low number K comes out and this suggest that a linear model would be good, that is why at the previous cross-validation also GLM is added as classifying algorithm. The optimum for classifying depends on the dataset, therefore, is the complete cross-validation and are all results run in this operator.

4.4.3. SMOTE & Balancing classes

To reduce the effect of the class imbalance problem, there are a few options. First will the results be tested with and without downsampling and SMOTE.

accuracy: 80.90% +/- 2.23% (mikro: 80.90%)

	true positive	true negative	true neutral	class precision
pred. positive	5526	560	610	82.53%
pred. negative	85	128	53	48.12%
pred. neutral	21	8	9	23.68%
class recall	98.12%	18.39%	1.34%	

Figure 26 - ML with GLM on HPC without sampling (SMOTE & downsampling)

Above: High accuracy but bad performance on the neutral and negative classes. Below: Lower accuracy as without sampling but better splitting over de classes and this will be used in the model.

accuracy: 67.23% +/- 4.10% (mikro: 67.23%)

	true negative	true positive	true neutral	class precision
pred. negative	253	125	142	48.65%
pred. positive	266	1327	388	66.99%
pred. neutral	177	237	1159	73.68%
class recall	36.35%	78.57%	68.62%	

Figure 27 - ML with GLM on HPC with sampling on (SMOTE & downsampling)

Target=HPC						
Algorithm=kNN	Features	Accur.	Error	Recall	Precision	Log loss
ML Training	no sampling	75.69%	24.31%	35.87%	38.31%	0.434
ML Testing	no sampling	75.20%	24.80%	35.13%	36.71%	0.435
ML Training	under sample	43.47%	56.53%	38.18%	37.14%	0.514
ML Testing	under sample	46.70%	53.30%	41.75%	38.47%	0.491
ML Training	under+oversample	84.89%	15.11%	65.73%	67.04%	0.372
ML Testing	under+oversample	60.40%	39.60%	38.48%	36.29%	0.467

Table 16 Sampling differences on HPC

4.4.4. Build models

Here are the RapidMiner processes created for the ML-model, Full-TL, and Combined-TL. An explanation of the operators, sequence, and parameters in appendix 1.

Roughly are the steps:

- Step 1 – Load data
- Step 2 – Pre-process data
- Step 3 – Split data (not in Full-TL model)
- Step 4 – Sampling
- Step 5 – Extract a vector from text
- Step 6 – Apply topic modeling
- Step 7 – Optimize Parameters/Cross-validation
- Step 8 – Testing or transferring

ML-model

First, the total ML-stage model is used to measure the model performance and contains the eight-steps like written in appendix 1.

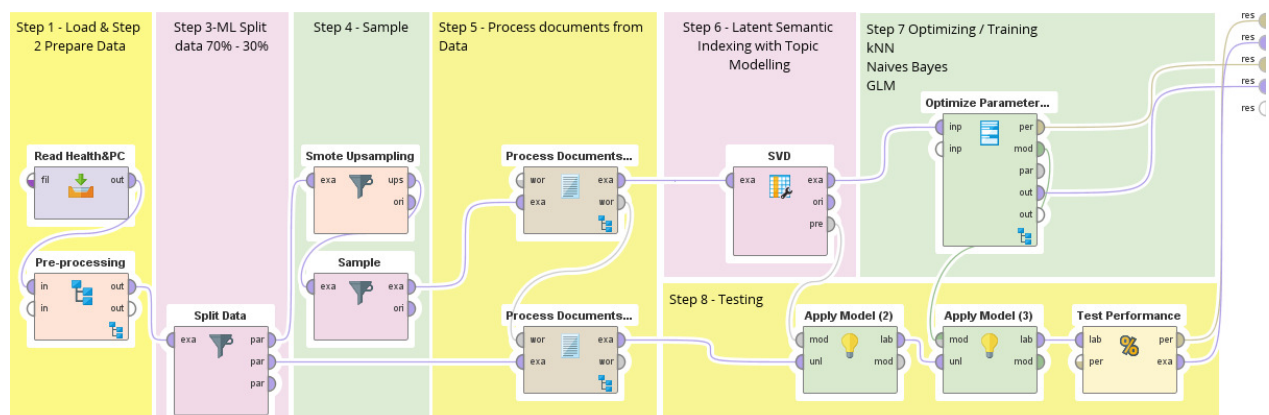


Figure 28 – ML model - Standard classifier with split data operator

Full-ML model

After the standard ML-model will run the Full-TL model. Here is no knowledge of the target reused and fully trained from the source.

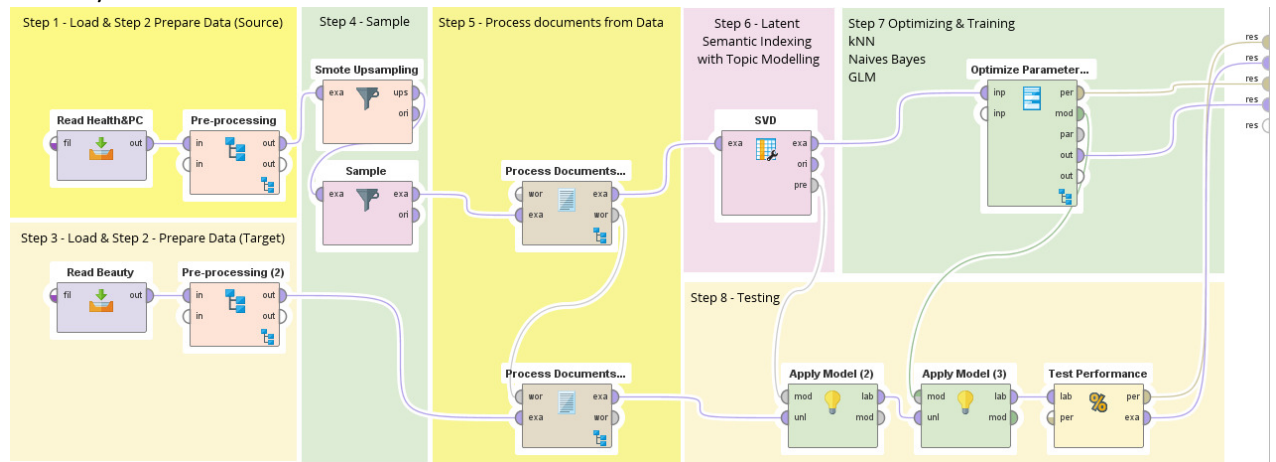


Figure 29 – Full-TL model

Combined-TL model

In this model is knowledge gathered from source/target combined to get optimal results from TL.

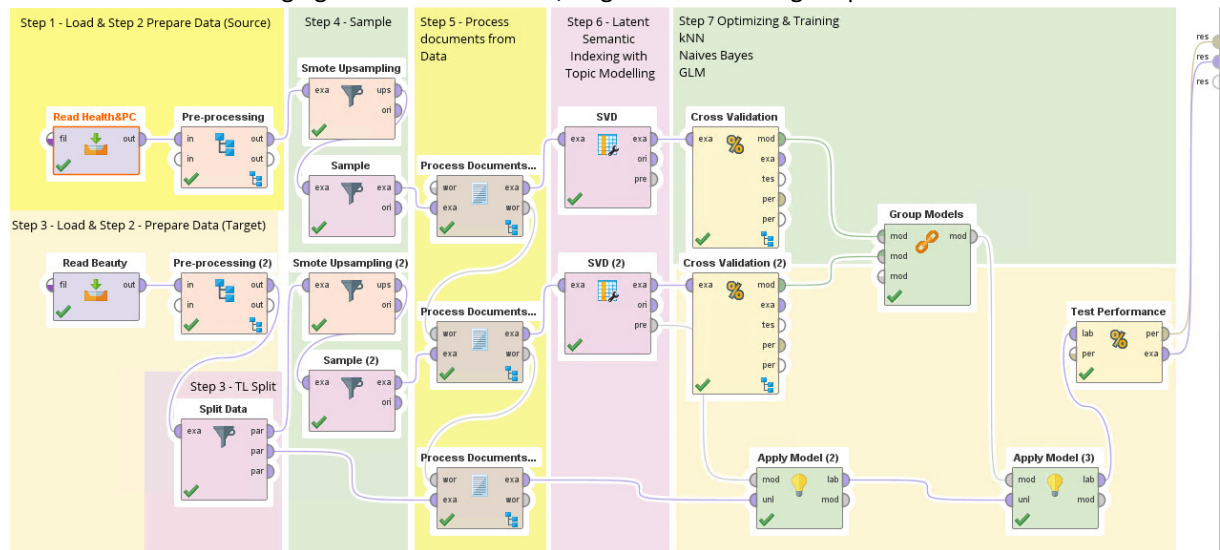


Figure 30 Combined-TL model

4.4.1. Asses models

The results will be placed in a table sorted by target, algorithm, and source. The first column represents the job, then accuracy, error, recall, precision, logistic loss, f1 score and confidence interval (the last few are not in the screenshot).

If the accuracy and recall is better than the **testing** of the ML-proces this is considered as an PT (green), if its worse then as a NT (red). If for instance the example above, if the accuracy is a PT and recall NT is is considered as neutral. The training details are added to detect overfitting.

	Accur	Error	Recall	Prec.	PT/NT?
ML Testing	60,23%	39,77%	45,14%	39,19%	
Full-TL source=Bea	51,94%	48,06%	41,23%	37,62%	[NT - NT]
Full-TL source=Auto	53,16%	46,84%	54,37%	49,14%	[PT - PT]
Com-TL source=Auto	59,73%	40,27%	44,11%	39,38%	[NT - PT]

4.5. CRISP-DM: Evaluation

In this paragraph are shown results of the different targets.

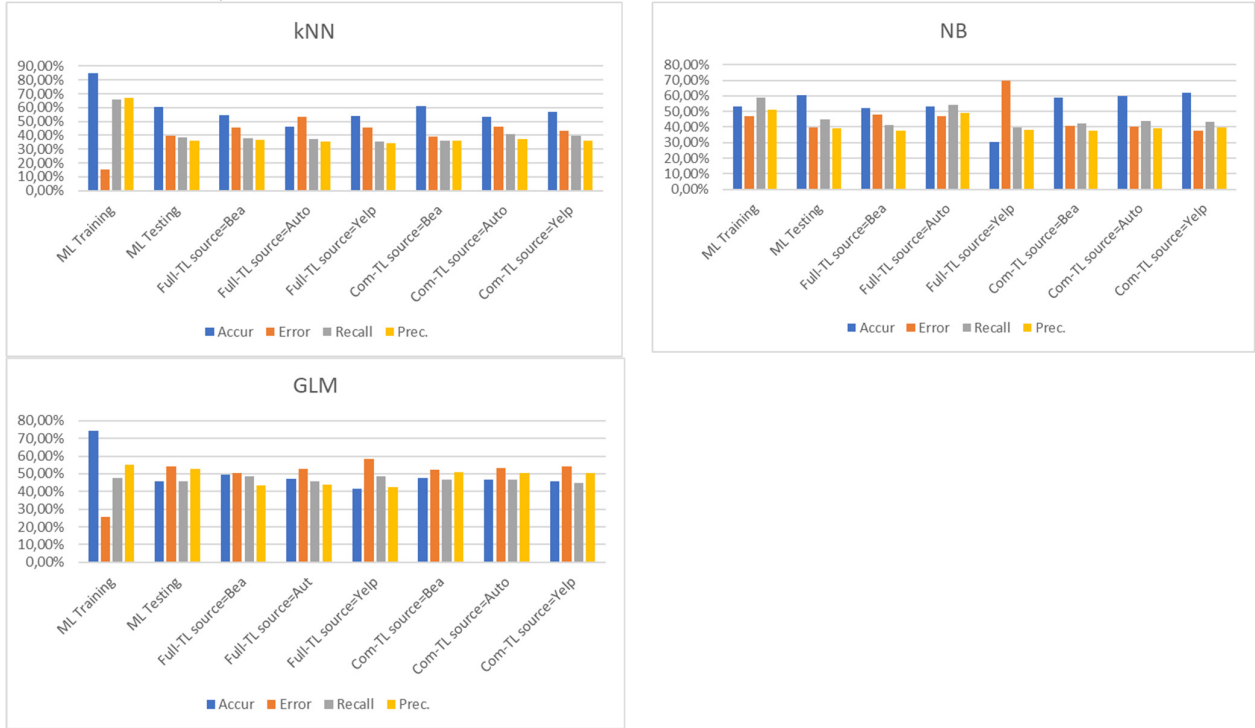
4.5.1. Target=HPC

Below are the results of the transfers where HPC is the target.

Target=HPC							
red=NT, green=PT							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	84,89%	15,11%	65,73%	67,04%	0,372	0,66379	[0,1433 - 0,1589]
ML Testing	60,40%	39,60%	38,48%	36,29%	0,467	0,37353	[0,3785 - 0,4135]
Full-TL source=Bea	54,33%	45,67%	37,64%	36,71%	0,487	0,37169	[0,4469 - 0,4665]
Full-TL source=Auto	46,34%	53,66%	37,22%	35,51%	0,517	0,36345	[0,5268 - 0,5464]
Full-TL source=Yelp	54,12%	45,88%	35,58%	34,13%	0,488	0,3484	[0,4490 - 0,4686]
Com-TL source=Bea	61,00%	39,00%	36,39%	36,03%	0,461	0,36209	[0,3725 - 0,4075]
Com-TL source=Auto	53,53%	46,47%	40,83%	37,17%	0,490	0,38914	[0,4469 - 0,4825]
Com-TL source=Yelp	56,83%	43,17%	39,54%	36,27%	0,477	0,37834	[0,4140 - 0,4494]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	53,18%	46,82%	58,94%	51,11%	0,490	0,54746	[0,4573 - 0,4791]
ML Testing	60,23%	39,77%	45,14%	39,19%	0,459	0,41955	[0,3802 - 0,4152]
Full-TL source=Bea	51,94%	48,06%	41,23%	37,62%	0,481	0,39342	[0,4708 - 0,4904]
Full-TL source=Auto	53,16%	46,84%	54,37%	49,14%	0,490	0,51623	[0,4586 - 0,4782]
Full-TL source=Yelp	30,19%	69,81%	39,63%	38,26%	0,561	0,38933	[0,6891 - 0,7071]
Com-TL source=Bea	59,13%	40,87%	42,14%	37,71%	0,462	0,39802	[0,3911 - 0,4263]
Com-TL source=Auto	59,73%	40,27%	44,11%	39,38%	0,462	0,41611	[0,3851 - 0,4203]
Com-TL source=Yelp	62,27%	37,73%	43,29%	39,93%	0,460	0,41542	[0,360 - 0,3946]
Algorithm=GLM	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	74,19%	25,81%	47,45%	55,30%	0,424	0,51075	[0,2485 - 0,2677]
ML Testing	45,70%	54,30%	45,87%	52,64%	0,505	0,49022	[0,5252 - 0,5608]
Full-TL source=Bea	49,62%	50,38%	48,39%	43,24%	0,504	0,4567	[0,4940 - 0,5136]
Full-TL source=Aut	47,13%	52,87%	45,54%	44,04%	0,505	0,44777	[0,5189 - 0,5385]
Full-TL source=Yelp	41,67%	58,33%	48,38%	42,45%	0,523	0,45221	[0,5736 - 0,5930]
Com-TL source=Bea	47,60%	52,40%	46,55%	50,94%	0,504	0,48646	[0,5061 - 0,5419]
Com-TL source=Auto	46,70%	53,30%	46,47%	50,48%	0,501	0,48392	[0,5151 - 0,5509]
Com-TL source=Yelp	45,90%	54,10%	44,72%	50,59%	0,506	0,47474	[0,5232 - 0,5588]
averages	54,16%	45,84%	44,73%	43,38%	0,476	0,44048	[0,4448 - 0,4721]
min	30,19%	15,11%	35,58%	34,13%	0,372	0,3484	[0,1433 - 0,1589]
max	84,89%	69,81%	65,73%	67,04%	0,523	0,66379	[0,6891 - 0,7071]

Algorithm performance

Below the visual representation of the results.



Meaningful confusion matrixes

These matrixes are used to see how the actual predictions have worked out.

accuracy: 61.00%

	true positive	true negative	true neutral	class precision
pred. positive	1718	192	173	82.48%
pred. negative	507	76	79	11.48%
pred. neutral	189	30	36	14.12%
class recall	71.17%	25.50%	12.50%	

Figure 31 Combined-TL Target=HPC, Source=Automotive, NB and leads to a PT

accuracy: 53.53%

	true negative	true positive	true neutral	class precision
pred. negative	158	792	120	14.77%
pred. positive	122	1417	137	84.55%
pred. neutral	18	205	31	12.20%
class recall	53.02%	58.70%	10.76%	

Figure 32 - Combined-TL Target=HPC, source=Automotive, kNN and leads to a PT

accuracy: 46.34%

	true positive	true negative	true neutral	class precision
pred. positive	4030	385	410	83.52%
pred. negative	2377	381	327	12.35%
pred. neutral	1639	228	223	10.67%
class recall	50.09%	38.33%	23.23%	

Figure 33 Combined-TL Target=HPC, Source=Beauty, kNN and leads to an NT

4.5.2. Target=Beauty

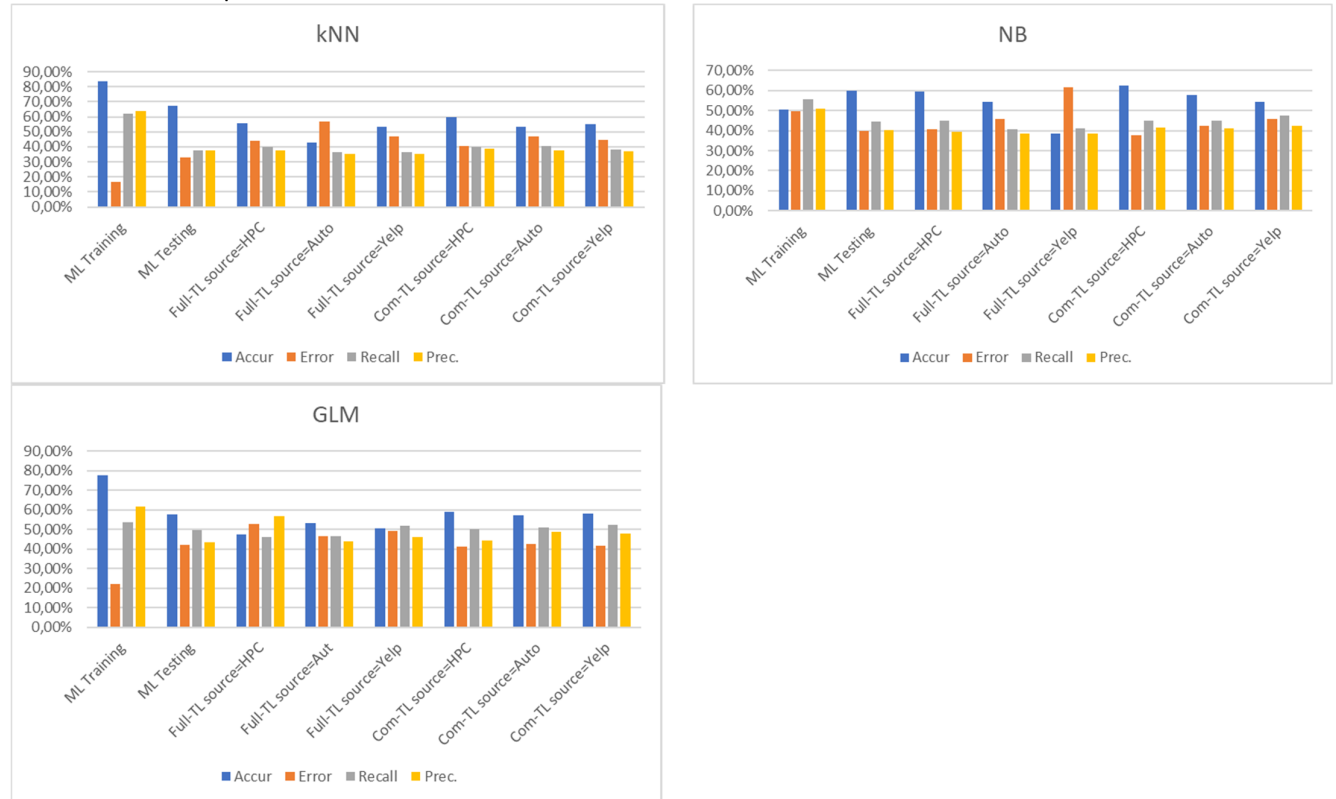
Below are the results of the transfers where Beauty is the target.

Target=Beauty							
red=NT, green=PT							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	83,63%	16,37%	62,02%	64,01%	0,377	0,62999	[0,1555 - 0,1719]
ML Testing	67,20%	32,80%	37,66%	37,77%	0,451	0,37715	[0,3112 - 0,3448]
Full-TL source=HPC	55,71%	44,29%	39,94%	37,68%	0,482	0,38777	[0,4332 - 0,4526]
Full-TL source=Auto	43,02%	56,98%	36,65%	35,43%	0,530	0,3603	[0,5601 - 0,5795]
Full-TL source=Yelp	53,22%	46,78%	36,50%	35,12%	0,491	0,35797	[0,4580 - 0,4776]
Com-TL source=HPC	59,60%	40,40%	40,03%	38,59%	0,467	0,39297	[0,3864 - 0,4216]
Com-TL source=Auto	53,17%	46,83%	40,22%	37,52%	0,491	0,38823	[0,4504 - 0,4862]
Com-TL source=Yelp	55,30%	44,70%	38,18%	36,99%	0,483	0,37576	[0,4292 - 0,4648]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	50,53%	49,47%	55,64%	50,91%	0,499	0,5317	[0,4837 - 0,5057]
ML Testing	59,97%	40,03%	44,43%	40,34%	0,463	0,42286	[0,3828 - 0,4178]
Full-TL source=HPC	59,26%	40,74%	44,98%	39,55%	0,464	0,42091	[0,3978 - 0,4170]
Full-TL source=Auto	54,24%	45,76%	40,80%	38,71%	0,474	0,39728	[0,4478 - 0,4674]
Full-TL source=Yelp	38,51%	61,49%	41,27%	38,73%	0,532	0,3996	[0,6054 - 0,6244]
Com-TL source=HPC	62,30%	37,70%	45,04%	41,70%	0,460	0,43306	[0,3597 - 0,3943]
Com-TL source=Auto	57,73%	42,27%	45,08%	41,00%	0,472	0,42943	[0,4050 - 0,4404]
Com-TL source=Yelp	54,37%	45,63%	47,63%	42,52%	0,488	0,4493	[0,4385 - 0,4741]
Algorithm=GLM	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	77,82%	22,18%	53,61%	61,62%	0,413	0,57337	[0,2126 - 0,2310]
ML Testing	57,70%	42,30%	49,87%	43,50%	0,483	0,46468	[0,4053 - 0,4407]
Full-TL source=HPC	47,33%	52,67%	46,23%	56,59%	0,498	0,50888	[0,5169 - 0,5365]
Full-TL source=Aut	53,23%	46,77%	46,66%	43,94%	0,492	0,45259	[0,4579 - 0,4775]
Full-TL source=Yelp	50,69%	49,31%	51,77%	45,95%	0,501	0,48687	[0,4833 - 0,5029]
Com-TL source=HPC	58,97%	41,03%	50,05%	44,35%	0,479	0,47028	[0,3927 - 0,4279]
Com-TL source=Auto	57,47%	42,53%	50,83%	48,81%	0,484	0,498	[0,4076 - 0,4430]
Com-TL source=Yelp	58,17%	41,83%	52,29%	47,89%	0,482	0,49993	[0,4006 - 0,4360]

averages	57,05%	42,95%	45,72%	43,72%	0,477	0,44698	[0,4159 - 0,4431]
min	38,51%	16,37%	36,50%	35,12%	0,377	0,35797	[0,1555 - 0,1719]
max	83,63%	61,49%	62,02%	64,01%	0,532	0,62999	[0,6054 - 0,6244]

Algorithm performance

Below the visual representation of the results.



Meaningful confusion matrixes

These matrixes are used to see how the actual predictions have worked out.

accuracy: 59.60%

	true positive	true negative	true neutral	class precision
pred. positive	1622	162	184	82.42%
pred. negative	258	72	51	18.90%
pred. neutral	472	85	94	14.44%
class recall	68.96%	22.57%	28.57%	

Figure 34 Combined-TL Target=Beauty, Source=HPC with kNN, PT

accuracy: 53.17%

	true positive	true negative	true neutral	class precision
pred. positive	1397	144	156	82.32%
pred. negative	487	114	89	16.52%
pred. neutral	468	61	84	13.70%
class recall	59.40%	35.74%	25.53%	

Figure 35 Combined-TL Target=Beauty, Source=Automotive with kNN, 'neutral'

accuracy: 55.30%

	true positive	true negative	true neutral	class precision
pred. positive	1493	154	179	81.76%
pred. negative	287	64	48	16.04%
pred. neutral	572	101	102	13.16%
class recall	63.48%	20.06%	31.00%	

Figure 36 Combined-TL Target=Beauty, Source=Automotive with kNN, 'neutral'

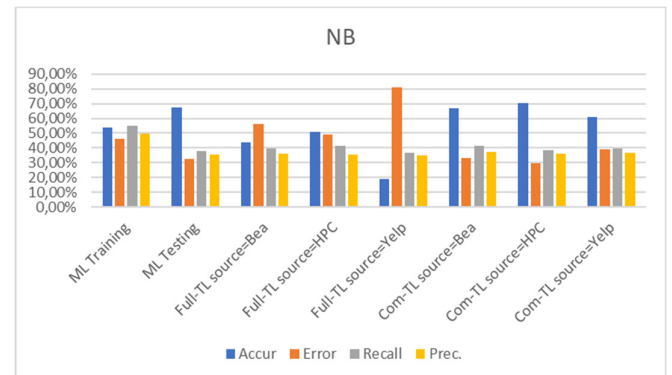
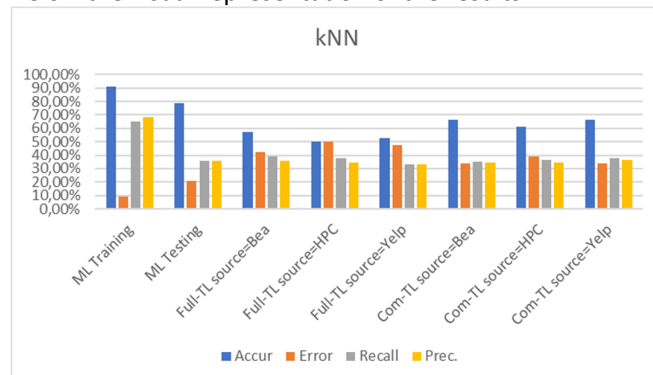
4.5.3. Target=Automotive

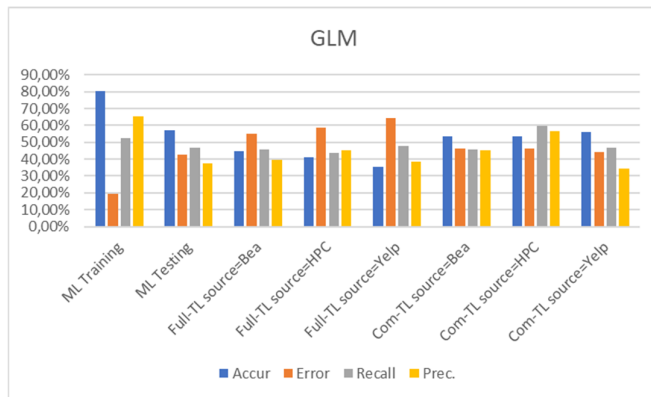
Below are the results of the transfers where Automotive is the target.

Target=Automotive							
red=NT, green=PT							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	91,04%	8,96%	64,83%	68,07%	0,354	0,66411	[0,0835 - 0,0957]
ML Testing	78,90%	21,10%	35,63%	35,96%	0,434	0,35794	[0,1964 - 0,2256]
Full-TL source=Bea	57,53%	42,47%	39,11%	36,10%	0,475	0,37545	[0,4150 - 0,4344]
Full-TL source=HPC	50,16%	49,84%	37,75%	34,75%	0,503	0,36188	[0,4886 - 0,5082]
Full-TL source=Yelp	52,52%	47,48%	33,47%	33,40%	0,494	0,33435	[0,4650 - 0,4846]
Com-TL source=Bea	66,13%	33,87%	35,12%	34,45%	0,442	0,34782	[0,3218 - 0,3556]
Com-TL source=HPC	61,20%	38,80%	36,19%	34,71%	0,461	0,35435	[0,3706 - 0,4054]
Com-TL source=Yelp	66,30%	33,70%	38,09%	36,29%	0,441	0,37168	[0,3201 - 0,3539]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	53,96%	46,04%	55,28%	49,66%	0,487	0,5232	[0,4498 - 0,4710]
ML Testing	67,23%	32,77%	37,64%	35,30%	0,436	0,36432	[0,3109 - 0,3445]
Full-TL source=Bea	43,78%	56,22%	39,81%	36,01%	0,505	0,37815	[0,5525 - 0,5719]
Full-TL source=HPC	51,00%	49,00%	41,51%	35,53%	0,486	0,38288	[0,4802 - 0,4998]
Full-TL source=Yelp	19,21%	80,79%	36,53%	34,92%	0,604	0,35707	[0,8002 - 0,8156]
Com-TL source=Bea	66,73%	33,27%	41,53%	37,47%	0,443	0,39396	[0,3158 - 0,3496]
Com-TL source=HPC	70,23%	29,77%	38,63%	36,30%	0,437	0,37429	[0,2813 - 0,3141]
Com-TL source=Yelp	61,10%	38,90%	39,46%	36,38%	0,459	0,37857	[0,3716 - 0,4064]
Algorithm=GLM	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	80,57%	19,43%	52,23%	65,35%	0,401	0,58058	[0,1859 - 0,2027]
ML Testing	57,17%	42,83%	47,02%	37,42%	0,480	0,41674	[0,4106 - 0,4460]
Full-TL source=Bea	44,98%	55,02%	45,83%	39,59%	0,518	0,42482	[0,5404 - 0,560]
Full-TL source=HPC	41,36%	58,64%	43,51%	45,19%	0,510	0,44334	[0,5767 - 0,5961]
Full-TL source=Yelp	35,65%	64,35%	47,67%	38,70%	0,539	0,42719	[0,6341 - 0,6529]
Com-TL source=Bea	53,70%	46,30%	45,90%	45,30%	0,489	0,45598	[0,4452 - 0,4808]
Com-TL source=HPC	53,58%	46,42%	59,50%	56,63%	0,491	0,5803	[0,4464 - 0,4820]
Com-TL source=Yelp	55,87%	44,13%	46,75%	34,57%	0,484	0,39748	[0,4235 - 0,4591]
averages	57,50%	42,50%	43,29%	40,75%	47,39%	0,41983	[0,4119 - 0,4382]
min	19,21%	8,96%	33,47%	33,40%	35,40%	0,33435	[0,0835 - 0,0957]
max	91,04%	80,79%	64,83%	68,07%	60,40%	0,66411	[0,8002 - 0,8156]

Algorithm performance

Below the visual representation of the results.





Meaningful confusion matrixes

These matrixes are used to see how the actual predictions have worked out.

accuracy: 66.13%

	true positive	true neutral	true negative	class precision
pred. positive	1920	143	112	88.28%
pred. neutral	446	44	40	8.30%
pred. negative	244	31	20	6.78%
class recall	73.56%	20.18%	11.63%	

Figure 37 Combined-TL Target=Automotive, Source=Beauty with kNN, NT

accuracy: 66.30%

	true positive	true neutral	true negative	class precision
pred. positive	1909	143	98	88.79%
pred. neutral	471	44	38	7.96%
pred. negative	230	31	36	12.12%
class recall	73.14%	20.18%	20.93%	

Figure 38 Combined-TL Target=Automotive, Source=Yelp with kNN, PT

accuracy: 55.30%

	true positive	true neutral	true negative	class precision
pred. positive	1517	71	31	93.70%
pred. neutral	15	3	2	15.00%
pred. negative	1078	144	139	10.21%
class recall	58.12%	1.38%	80.81%	

Figure 39 Combined-TL Target=Automotive, Source=Yelp with GLM, NT

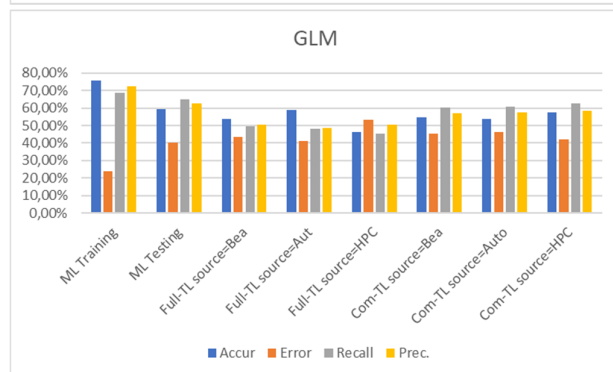
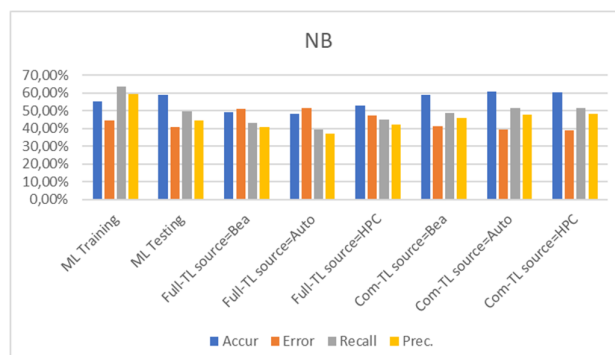
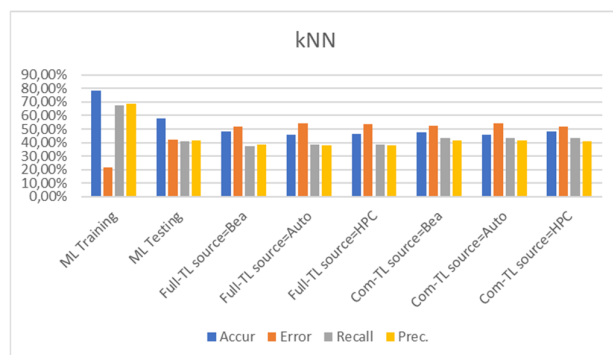
4.5.4. Target=Yelp

Below are the results of the transfers where Yelp is the target.

Target=Yelp							
<i>red=NT, green=PT</i>							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	78,34%	21,66%	67,63%	68,91%	0,404	0,68264	[0,2073 - 0,2259]
ML Testing	57,92%	42,08%	40,94%	41,50%	0,484	0,41218	[0,4031 - 0,4385]
Full-TL source=Bea	48,08%	51,92%	37,23%	38,53%	0,510	0,37869	[0,5094 - 0,5290]
Full-TL source=Auto	45,56%	54,44%	38,35%	37,63%	0,520	0,37987	[0,5346 - 0,5542]
Full-TL source=HPC	46,35%	53,65%	38,75%	38,13%	0,517	0,38438	[0,5267 - 0,5463]
Com-TL source=Bea	47,69%	52,31%	43,19%	41,42%	0,512	0,42286	[0,5052 - 0,5410]
Com-TL source=Auto	45,69%	54,31%	43,41%	41,46%	0,520	0,42413	[0,5253 - 0,5609]
Com-TL source=HPC	48,06%	51,94%	43,11%	41,12%	0,511	0,42091	[0,5015 - 0,5373]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	55,33%	44,67%	63,61%	59,57%	0,481	0,61524	[0,4355 - 0,4579]
ML Testing	59,09%	40,91%	49,60%	44,57%	0,464	0,46951	[0,3915 - 0,4267]
Full-TL source=Bea	49,16%	50,84%	43,32%	40,86%	0,503	0,42054	[0,4986 - 0,5182]
Full-TL source=Auto	48,45%	51,55%	39,57%	37,16%	0,500	0,38327	[0,5057 - 0,5253]
Full-TL source=HPC	52,83%	47,17%	45,17%	42,16%	0,482	0,43613	[0,4619 - 0,4815]
Com-TL source=Bea	58,86%	41,14%	48,68%	45,73%	0,465	0,47159	[0,3938 - 0,4290]
Com-TL source=Auto	60,59%	39,41%	51,29%	47,90%	0,460	0,49537	[0,3766 - 0,4116]
Com-TL source=HPC	60,13%	38,87%	51,29%	48,36%	0,461	0,49782	[0,3713 - 0,4061]
Algorithm=GLM	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	75,95%	24,05%	68,81%	72,60%	0,418	0,70654	[0,2309 - 0,2501]
ML Testing	59,59%	40,41%	64,80%	62,83%	0,472	0,638	[0,3865 - 0,4217]
Full-TL source=Bea	53,63%	43,37%	49,38%	50,59%	0,485	0,49978	[0,4240 - 0,4434]
Full-TL source=Aut	58,78%	41,22%	47,98%	48,57%	0,477	0,48273	[0,4025 - 0,4219]
Full-TL source=HPC	46,46%	53,54%	45,23%	50,73%	0,503	0,47822	[0,5256 - 0,5452]
Com-TL source=Bea	54,68%	45,32%	60,46%	56,87%	0,486	0,5861	[0,4354 - 0,4710]
Com-TL source=Auto	53,78%	46,22%	60,82%	57,35%	0,493	0,59034	[0,4444 - 0,480]
Com-TL source=HPC	57,69%	42,31%	62,46%	58,27%	0,480	0,60292	[0,4054 - 0,4408]
averages	55,11%	44,72%	50,21%	48,87%	0,48367	0,4953	[0,4334 - 0,4610]
min	45,56%	21,66%	37,23%	37,16%	0,404	0,37195	[0,2073 - 0,2259]
max	78,34%	54,44%	68,81%	72,60%	0,52	0,70654	[0,5346 - 0,5609]

Algorithm performance

Below the visual representation of the results.



Meaningful confusion matrixes

These matrixes are used to see how the actual predictions have worked out.

accuracy: 47.69%

	true positive	true neutral	true negative	class precision
pred. positive	926	104	160	77.82%
pred. neutral	249	58	72	15.30%
pred. negative	804	176	443	31.13%
class recall	46.79%	17.16%	65.63%	

Figure 40 Combined-TL Target=Yelp, Source=Beauty with kNN, PT

accuracy: 54.68%

	true positive	true neutral	true negative	class precision
pred. positive	1003	42	41	92.36%
pred. neutral	747	250	251	20.03%
pred. negative	229	46	383	58.21%
class recall	50.68%	73.96%	56.74%	

Figure 41 Combined-TL Target=Yelp, Source=Beauty with GLM, NT

accuracy: 57.69%

	true positive	true negative	true neutral	class precision
pred. positive	1063	40	42	92.84%
pred. negative	212	423	56	61.22%
pred. neutral	704	212	240	20.76%
class recall	53.71%	62.67%	71.01%	

Figure 42 Combined-TL Target=Yelp, Source=HPC with GLM, NT

4.5.5. Evaluate results

Visual representation Full-TL or Combined-TL

Below are the results of the Full-TL and the Combined-TL model. The combined-TL model performs significantly better with 45% of PT and 22% of NT.

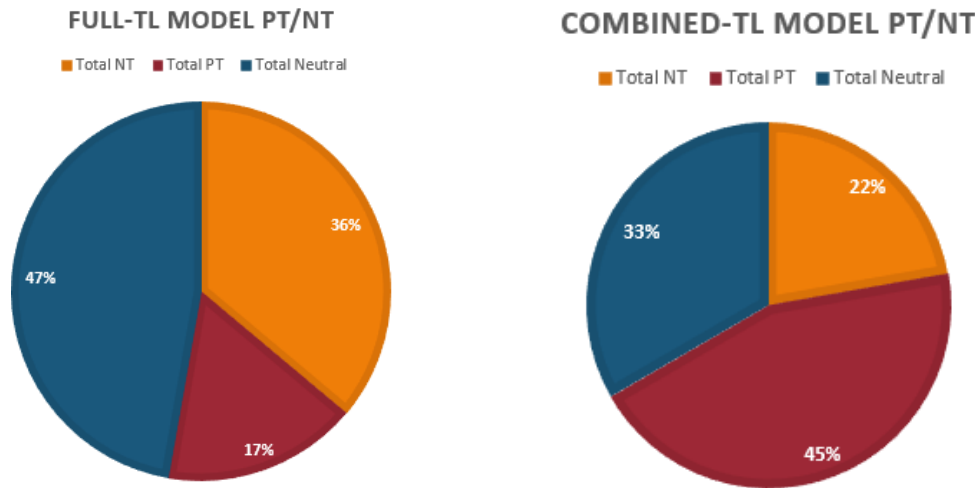


Figure 43 Results of the total Full-TL causes more NT and neutral transfers.

Because of the servant performance (only 17% of PT) on Full-TL are in the following pie-charts only the Combined-TL model results included.

Algorithms

There are differences found in the transfers per algorithm. Naïve Bayes produces the best results with only 8% of NT and 50% of PT. GLM the worst with 42% of NT and 33% of PT. There can be no parameters set in GLM, and it is a linear model, in this experiment this does not work that good with TL in this experiment.

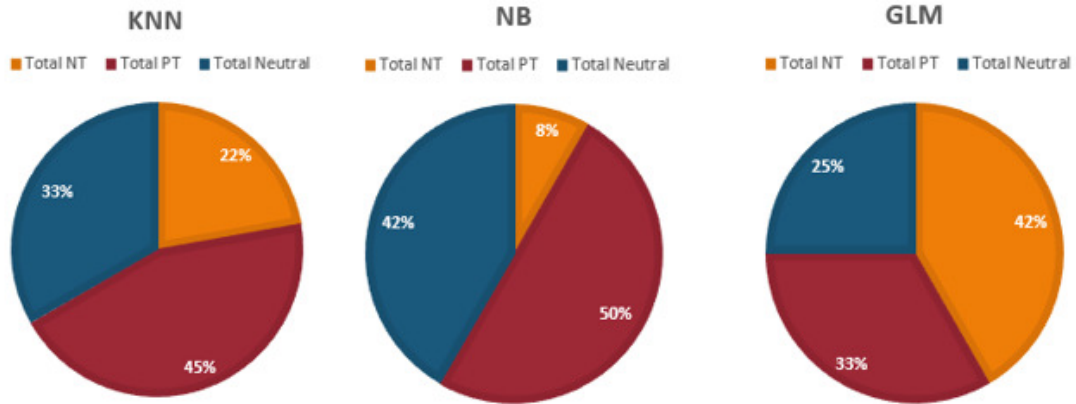


Figure 44 Naive Bayes causes te lowest percentage of NT, GLM the highest.

Targets

TL uses a target, to learn from and source, to transfer too. On the target side, transfers to Beauty work well with 78% of PT and no NT. Only transfers from Automotive and Yelp perform worse with neutral transfers. Yelp is contrary and uses entirely different words in the distribution (services instead of products).

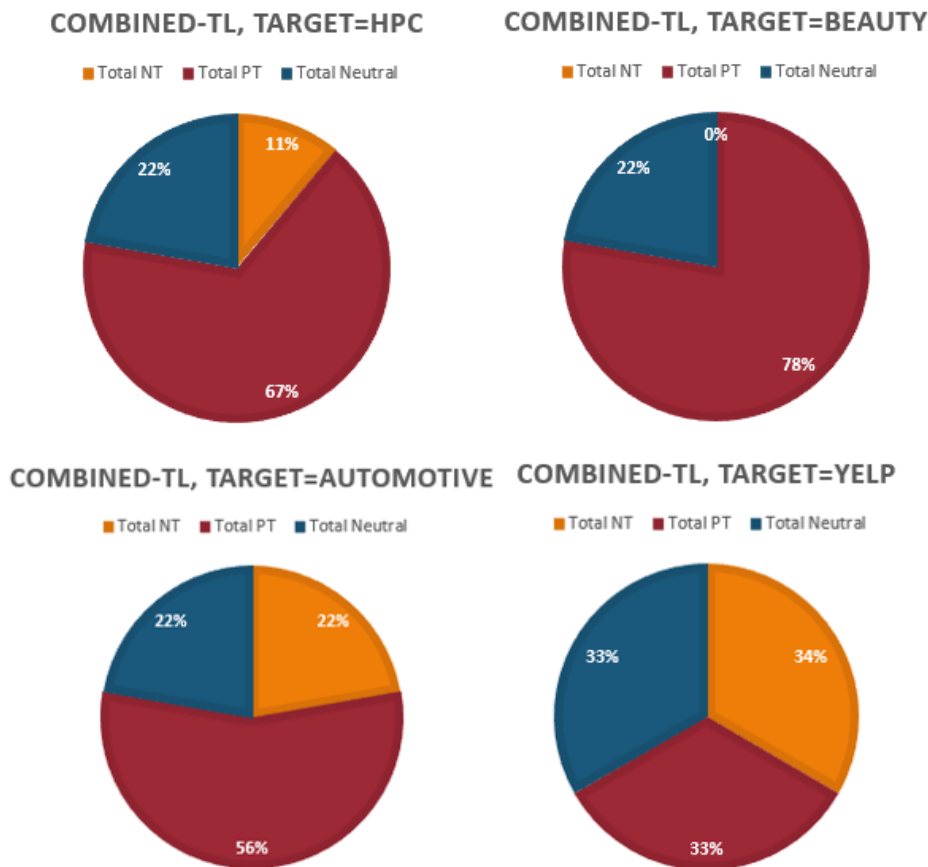


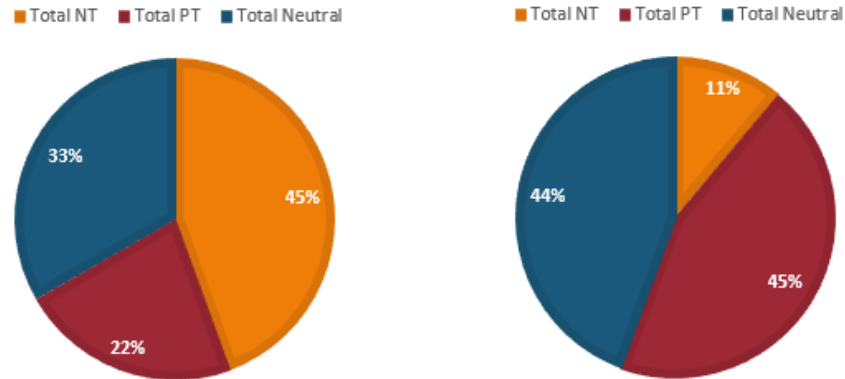
Figure 45 There is no NT when Beauty is the target. Transfer to Yelp causes the most NT.

An evident result is that all transfers to where target=Yelp and GLM is the algorithm are all NT. Also, can be seen that the NB algorithm performs with a higher recall but lower precision. kNN results with three PT in the same dataset, so it has fewer problems to create a good line to make the predictions.

Sources

When Beauty is used as source it causes 45% of NT, in contrary 0% NT when used as a target. HPC and Automotive have the lowest NT ratio.

COMBINEDTL, SOURCE=BEAUTY COMBINEDTL, SOURCE=AUTOMOTIVE



COMBINEDTL, SOURCE=YELP COMBINEDTL, SOURCE=HPC

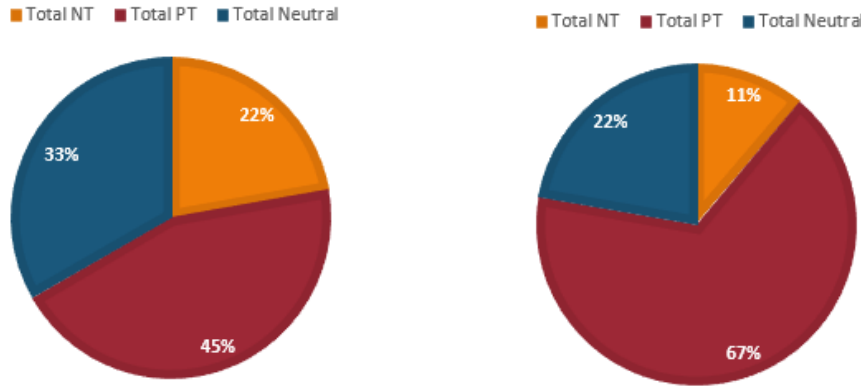


Figure 46 Automotive/HPC causes the lowest NT. Beauty the highest percentage of NT.

4.5.1. Extra experiment 1 – Deep Learning & Neural Nets

Because during the elaboration of the empirical research knowledge is gathered that deep learning can have improvement on the TL is done a small experiment on the best Combined-TL model that can be used for further research. Therefore, the algorithms Deep Learning algorithms H2O and Neural Net are tested.

Target=HPC							
Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	77,26%	22,74%	63,75%	62,67%	0,408	0,63205	[0,2182 - 0,2366]
ML Testing	69,67%	30,33%	47,31%	44,64%	0,44	0,45936	[0,2869 - 0,3197]
Com-TL source=Bea	61,83%	38,17%	47,23%	42,10%	0,463	0,44518	[0,3643 - 0,3991]
Com-TL source=Auto	70,40%	29,60%	49,77%	45,62%	0,439	0,47605	[0,2797 - 0,3123]
Com-TL source=Yelp	73,23%	26,77%	46,45%	45,53%	0,43	0,45985	[0,2519 - 0,2835]
Neutral Nets	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	87,51%	12,49%	68,87%	73,42%	0,361	0,71072	[0,1177 - 0,1321]
ML Testing	62,93%	37,07%	45,33%	40,84%	0,452	0,42968	[0,3534 - 0,3880]
Com-TL source=Bea	64,43%	35,57%	45,05%	41,36%	0,446	0,43126	[0,3386 - 0,3728]
Com-TL source=Auto	65,53%	34,47%	49,06%	43,71%	0,444	0,46231	[0,3277 - 0,3617]
Com-TL source=Yelp	62,53%	37,47%	44,73%	40,74%	0,455	0,42642	[0,3574 - 0,3920]

accuracy: 62.53%

	true positive	true neutral	true negative	class precision
pred. positive	1687	145	118	86.51%
pred. neutral	356	75	66	15.09%
pred. negative	371	68	114	20.61%
class recall	69.88%	26.04%	38.26%	

Figure 43 Combined-TL from HPC to Yelp with Neural Net, a NT

Target=Beauty							
Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	76,78%	23,22%	63,71%	61,49%	0,408	0,6258	[0,2229 - 0,2415]
ML Testing	69,10%	30,90%	48,23%	44,46%	0,442	0,46268	[0,2925 - 0,3255]
Com-TL source=HPC	68,33%	31,67%	49,82%	48,56%	0,443	0,49182	[0,3001 - 0,3333]
Com-TL source=Auto	66,27%	33,73%	48,86%	44,75%	0,446	0,46715	[0,3204 - 0,3542]
Com-TL source=Yelp	64,20%	35,80%	52,98%	45,94%	0,458	0,49209	[0,3408 - 0,3752]
Neutral Nets	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	83,37%	13,63%	66,43%	70,39%	0,364	0,68353	[0,1287 - 0,1439]
ML Testing	64,80%	35,20%	49,66%	45,28%	0,446	0,47369	[0,3349 - 0,3691]
Com-TL source=HPC	65,17%	34,83%	48,25%	44,35%	0,445	0,46218	[0,3313 - 0,3653]
Com-TL source=Auto	62,23%	37,77%	47,47%	43,12%	0,455	0,45191	[0,3604 - 0,3950]
Com-TL source=Yelp	64,43%	35,57%	49,35%	44,66%	0,446	0,46888	[0,3386 - 0,3728]

Target=Automotive							
Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	83,04%	16,96%	63,90%	63,44%	0,391	0,63669	[0,1616 - 0,1776]
ML Testing	84,10%	15,90%	40,22%	44,07%	0,403	0,42057	[0,1459 - 0,1721]
Com-TL source=Bea	79,83%	20,17%	43,11%	42,45%	0,421	0,42777	[0,1873 - 0,2161]
Com-TL source=HPC	84,53%	15,47%	42,47%	47,01%	0,405	0,44625	[0,1418 - 0,1676]
Com-TL source=Yelp	82,87%	17,13%	39,27%	41,09%	0,404	0,40159	[0,1578 - 0,1848]
Neutral Nets	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	91,15%	8,85%	69,08%	71,07%	0,346	0,70061	[0,0824 - 0,0946]
ML Testing	71,70%	28,30%	43,15%	40,48%	0,419	0,41772	[0,2669 - 0,2991]
Com-TL source=Bea	69,57%	30,43%	44,39%	39,42%	0,428	0,41758	[0,2878 - 0,3208]
Com-TL source=HPC	72,13%	27,87%	44,29%	39,92%	0,418	0,41992	[0,2627 - 0,2947]
Com-TL source=Yelp	69,57%	30,43%	41,71%	38,40%	0,427	0,39987	[0,2878 - 0,3208]

accuracy: 69.57%

	true positive	true neutral	true negative	class precision
pred. positive	1973	119	88	90.50%
pred. neutral	395	71	41	14.00%
pred. negative	242	28	43	13.74%
class recall	75.59%	32.57%	25.00%	

Figure 44 Combined-TL from Automotive to Yelp with Neural Net, a NT

Target=Yelp							
Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	81,17%	18,83%	77,11%	76,40%	0,391	0,76753	[0,1795 - 0,1971]
ML Testing	65,47%	35,53%	59,55%	54,66%	0,445	0,57	[0,3382 - 0,3724]
Com-TL source=HPC	63,44%	36,56%	56,78%	52,74%	0,452	0,54685	[0,3484 - 0,3828]
Com-TL source=Auto	60,70%	39,30%	56,43%	51,75%	0,462	0,53989	[0,3755 - 0,4105]
Com-TL source=Bea	65,81%	34,19%	56,56%	52,98%	0,446	0,54711	[0,3249 - 0,3589]
Neutral Nets	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	87,28%	12,72%	80,11%	84,19%	0,361	0,82099	[0,1197 - 0,1347]
ML Testing	68,01%	31,99%	58,70%	54,69%	0,434	0,56624	[0,3032 - 0,3366]
Com-TL source=HPC	61,73%	38,27%	55,61%	51,84%	0,457	0,53659	[0,3653 - 0,4001]
Com-TL source=Auto	57,39%	42,61%	54,14%	50,31%	0,471	0,52155	[0,4084 - 0,4438]
Com-TL source=Bea	60,63%	39,37%	54,81%	50,81%	0,461	0,52734	[0,3762 - 0,4112]

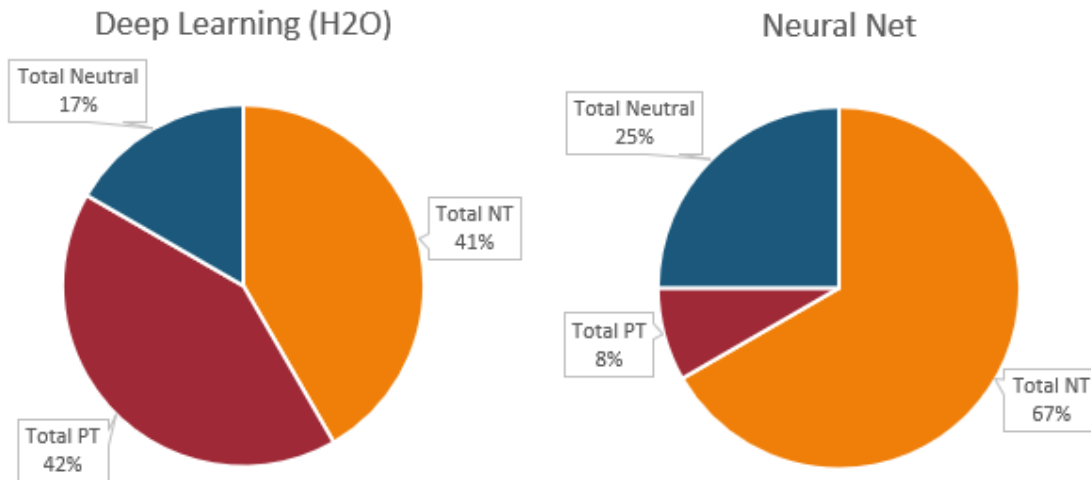


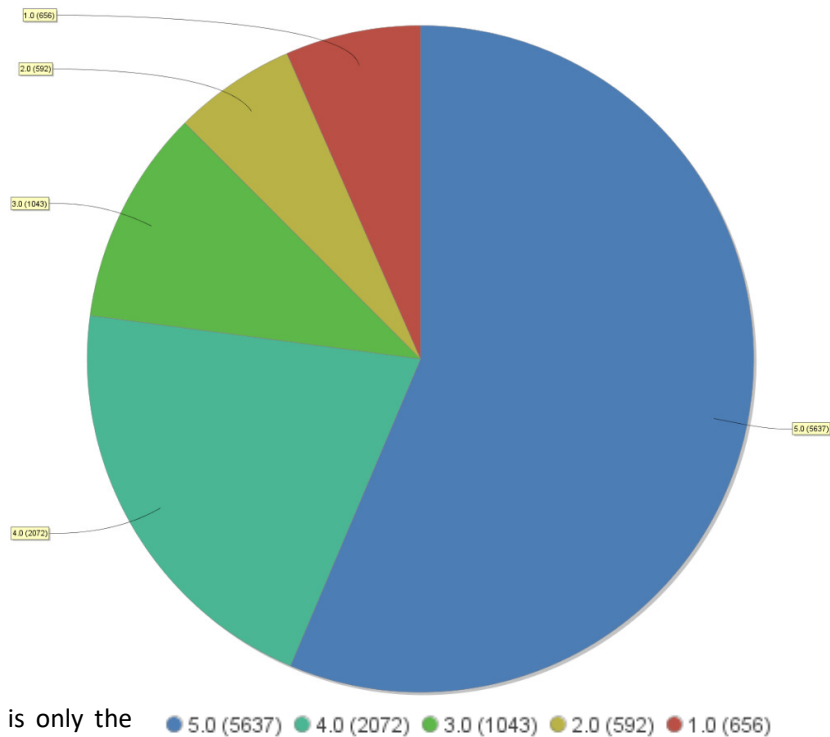
Figure 45 Neural Net causes the most NT, in comparison to all other algorithms

As can be seen in the tables before deep learning (H2O) and neural net (multi-layer-perceptron) get good results on the ML-model but have lower rates on the TL-model. Neural Net performs the worst with 67% of NT. When Yelp is the target everything is NT.

4.5.2. Extra experiment 2 – Add extra datasets

CPA dataset

The Amazon Cell Phone and Accessories (CPA) review dataset is Amazon's most balanced review dataset but still imbalanced (He & McAuley, 2016) and this is useful to see because this are product reviews of a different category. The data is stored on the same way as within the other Amazon datasets, so the processes could be reused. Because of the experiences written in 4.4.5 will only the Combined-TL model + kNN, NB and Deep Learning (H2O) be presented.



In the table can be seen that all here are all NB NT and kNN are PT, on H2O is only the transfer to HPC neutral, the rest are PT's.

Figure 46 CPA also imbalanced, examples per class

Target=CPA							
red=NT, green=PT							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	86,95%	13,05%	70,87%	72,74%	0,363	0,71792825	[0,1231 - 0,1379]
ML Testing	60,53%	39,47%	39,04%	37,41%	0,463	0,382076233	[0,3772 - 0,4122]
Com-TL source=HPC	55,70%	44,30%	40,77%	38,75%	0,482	0,397343436	[0,4252 - 0,4608]
Com-TL source=Bea	51,47%	48,53%	40,25%	37,62%	0,498	0,388905869	[0,4674 - 0,5032]
Com-TL source=Auto	52,60%	47,40%	42,00%	38,76%	0,493	0,403150074	[0,4561 - 0,4919]
Com-TL source=Yelp	54,83%	45,17%	43,18%	39,82%	0,485	0,414319904	[0,4339 - 0,4695]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	53,40%	46,60%	61,18%	53,09%	0,49	0,568486252	[0,4551 - 0,4769]
ML Testing	65,80%	34,20%	47,19%	42,52%	0,445	0,447334478	[0,3250 - 0,3590]
Com-TL source=HPC	54,07%	45,93%	46,41%	41,50%	0,482	0,438178819	[0,4415 - 0,4771]
Com-TL source=Bea	52,93%	47,07%	46,11%	41,78%	0,488	0,438383388	[0,4528 - 0,4886]
Com-TL source=Auto	50,80%	49,20%	46,72%	41,95%	0,495	0,44206699	[0,4741 - 0,5099]
Com-TL source=Yelp	52,17%	47,83%	45,32%	42,13%	0,491	0,436668176	[0,4604 - 0,4962]
Algorithm=Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	77,92%	22,08%	69,58%	66,37%	0,404	0,679371033	[0,2117 - 0,2299]
ML Testing	63,77%	36,23%	51,62%	45,24%	0,455	0,482198802	[0,3451 - 0,3795]
Com-TL source=HPC	60,03%	39,97%	51,27%	45,78%	0,469	0,483697187	[0,3822 - 0,4172]
Com-TL source=Bea	62,87%	37,13%	53,91%	47,00%	0,462	0,502184124	[0,3540 - 0,3886]
Com-TL source=Auto	63,53%	36,47%	51,81%	46,03%	0,460	0,487492702	[0,3475 - 0,3819]
Com-TL source=Yelp	64,83%	35,17%	52,20%	46,62%	0,453	0,49252459	[0,3346 - 0,3688]
averages	57,60%	42,40%	47,42%	44,01%	0,473	0,456491681	[0,4077 - 0,4403]
min	50,80%	13,05%	39,04%	37,41%	0,363	0,382076233	[0,1231 - 0,1379]
max	86,95%	49,20%	70,87%	72,74%	0,498	0,71792825	[0,4741 - 0,5099]

The results of the CPA dataset as target are printed below, very interesting is that the best performing algorithm overall scores here the worst.

COMBINED-TL, TARGET=CPA

■ Total NT ■ Total PT ■ Total Neutral

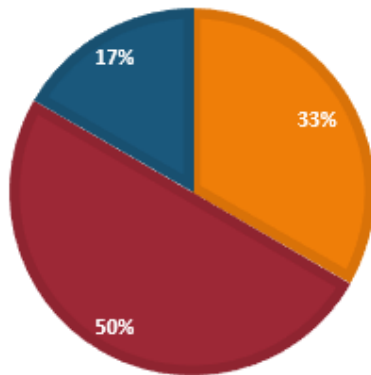


Figure 47 Despite of the better-balanced dataset Amazon CPA causes relative high number of NT.

Edmunds Car review database

This is a dataset collected from the Edmunds Car review database and contain 598 car reviews with a grading system which is also converted to positive, neutral or negative class.

First is the data with the rating is stored in CSV files and the reviewText in files per year, so this should be combined and stored in a new CSV file.

Then is the Combined-TL model slightly changed because this csv file is already tokenized.

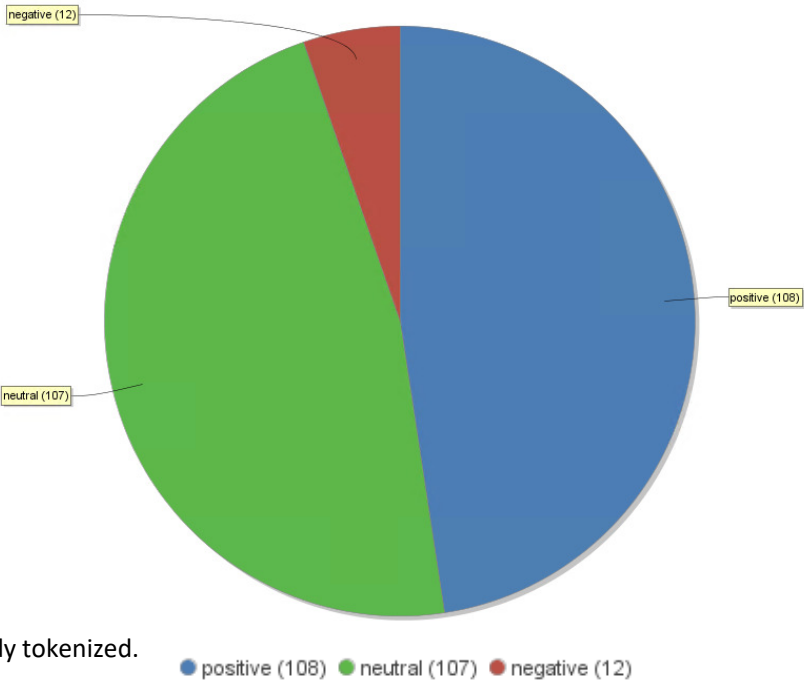


Figure 48 Imbalanced Edmunds dataset

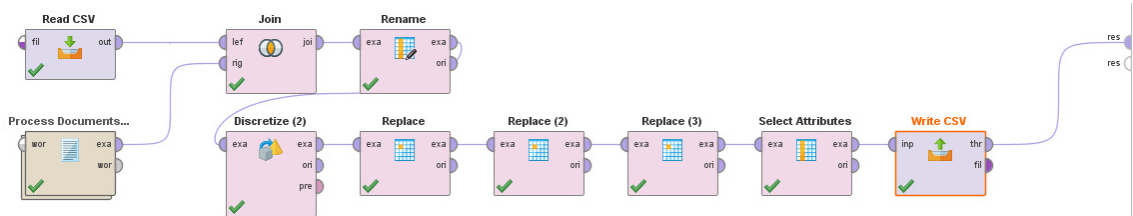


Figure 49 Preprocessing “Combining rating with text”

accuracy: 39.44%

	true neutral	true negative	true positive	class precision
pred. neutral	48	6	72	38.10%
pred. negative	10	2	8	10.00%
pred. positive	12	1	21	61.76%
class recall	68.57%	22.22%	20.79%	

Figure 50 Transfer with deep learning from Automotive to Edwards, an NT

Target=Edwards							
red=NT, green=PT							
Algorithm=kNN	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	50,83%	49,17%	41,83%	39,10%	0,5	0,404189547	[0,4917 - 0,4917]
ML Testing	46,11%	53,89%	38,66%	43,18%	0,518	0,407951808	[0,5389 - 0,5389]
Com-TL source=HPC	34,44%	65,56%	45,23%	43,67%	0,562	0,444363127	[0,6556 - 0,6556]
Com-TL source=Bea	35,56%	64,44%	40,02%	37,23%	0,558	0,385746175	[0,6444 - 0,6444]
Com-TL source=Auto	29,44%	70,56%	28,77%	36,42%	0,581	0,32146139	[0,7056 - 0,7056]
Com-TL source=Yelp	35,00%	65,00%	33,39%	41,36%	0,560	0,36950111	[0,650 - 0,650]
Com-TL source=CPA	37,78%	62,22%	51,03%	38,87%	0,550	0,441276107	[0,6222 - 0,6222]
Algorithm=NB	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	51,58%	48,42%	33,57%	32,73%	0,492	0,331446787	[0,4842 - 0,4842]
ML Testing	48,89%	51,11%	46,33%	45,51%	0,501	0,459163393	[0,5111 - 0,5111]
Com-TL source=HPC	41,67%	58,33%	43,80%	38,35%	0,525	0,408942179	[0,5833 - 0,5833]
Com-TL source=Bea	44,44%	55,56%	46,77%	42,62%	0,521	0,445986665	[0,5556 - 0,5556]
Com-TL source=Auto	48,33%	51,67%	43,50%	44,07%	0,509	0,437831449	[0,5167 - 0,5167]
Com-TL source=Yelp	38,89%	61,11%	35,70%	33,55%	0,537	0,345916245	[0,6111 - 0,6111]
Com-TL source=CPA	45,00%	55,00%	40,20%	40,31%	0,516	0,402549249	[0,550 - 0,550]
Algorithm=Deep learning (H2O)	Accur	Error	Recall	Prec.	L. Loss	F1 score	Confidence int.
ML Training	53,91%	46,09%	41,94%	44,93%	0,485	0,433835432	[0,4609 - 0,4609]
ML Testing	47,22%	52,78%	39,61%	41,92%	0,509	0,407322752	[0,5278 - 0,5278]
Com-TL source=HPC	46,11%	53,89%	47,46%	44,16%	0,511	0,457505697	[0,5389 - 0,5389]
Com-TL source=Bea	50,00%	5,00%	49,34%	45,07%	0,498	0,471084377	[0,050 - 0,050]
Com-TL source=Auto	39,44%	60,56%	37,20%	36,62%	0,529	0,369077215	[0,6056 - 0,6056]
Com-TL source=Yelp	41,11%	58,89%	35,40%	35,28%	0,529	0,353398981	[0,5889 - 0,5889]
Com-TL source=CPA	42,22%	57,78%	40,61%	36,94%	0,521	0,386881599	[0,5778 - 0,5778]
averages	42,00%	58,00%	40,63%	39,78%	0,531	0,402016316	[0,580 - 0,580]
min	29,44%	48,42%	28,77%	32,73%	0,492	0,306225073	[0,4842 - 0,4842]
max	51,58%	70,56%	51,03%	45,51%	0,581	0,481121877	[0,7056 - 0,7056]

COMBINED-TL, TARGET=EDWARDS

■ Total PT ■ Total NT ■ Total Neutral

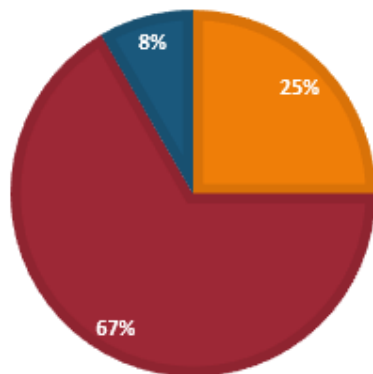


Figure 51 The Edwards dataset causes a high number of NT.

5. Conclusion, discussion, and recommendations, reflection

5.1. Conclusion

5.1.1. Q6. When is Transfer Learning most effective?

From the experiment can be concluded that MSTL (combined-TL) is, in this situation, the most effective with, 45% of PT, 33% neutral and 22% of NT. The Domain Adaptation TL model (full-TL) in this experiment causes 17% of PT, 36% neutral and 36% of NT. So, MSTL performs significant better. Therefore, are in the following totals the full-TL not included.

From an algorithm perspective, 50% of the NB transfers lead to a PT and, kNN 45% and GLM leads to only 33% of PT. The high results of NB can not be confirmed in the extra datasets. For instance on Amazon CPA all NB transfers have no PT and the Edwards have some more only 25% of PT. The extra algorithm results deep learning (H2O) causes 42% of PT and neural net only 8% of PT (!). Overall NB performs the best.

On the targetside are HPC 67%, Edwards 67% and Beauty 78% lead to the most PT. Yelp and Automotive have the lowest PT. HPC and Beauty have also the lowest neutral transfers. From a source prespective models that learn from HPC performs really well with 67% of PT, learning from Yelp and Automotive both 45% of PT, but because the neutral class of automotive is larger is automotive slightly better. Beauty with 22% of PT performs the worst, this is interessting because as target this is performing the best. The dataset similarity significance between datasets is small but can be seen between HPC and Beauty which both contain body products similar, just like the the WordNet comparisation in paragraph 4.1.3. This effect can not be seen between Amazon Automotive and Edwards Carreviews.

LSI Topic modelling causes a performance boost in the model, especially pre-model is created on the target-training, instead of source dataset, the results improve for another few percents.

5.1.2. Q7. How to avoid Negative Transfer?

This question is the contrary from Q6 so the answer have similarities. Chosing the right TL technique can have a huge impact on the NT. The domain adaption/full-TL model causes 36% of NT and 47% neutral transfers. The combined-TL model (MSTL) causes only 22% of NT and 33% of neutral transfers.

Focussed on algorithms, GLM causes the largest number of NT 42%, kNN 22% and NB only 8%. In the extra experiment can be seen at deep learning (H2O) casues 41% of NT and neural net 67% of NT.

Zoomed-in on the targetside can be seen that transfers to Beauty work well with 0% of NT, second best is HPC 11%, Automotive 22% and Yelp 34% of NT. From a source perspective HPC and Automotive have both 11%, Yelp 22% and Beauty 45% of NT. So there is a small significance on similarity of the word distribution. Also here topic modelling improves the results and reduces the number of NT.

5.1.3. Main research question

This research found a way to work with TL effectively and how the NT could be avoided and the central research question of this thesis was:

How to use Transfer Learning effectively and which factors cause a Negative Transfer?

This research showed that TL works well if MSTL is applied in combination with the right algorithm and, topic modeling (LSI) improves the results. The higher the similarity of the distribution of words from source and target the better TL functions.

NT is contrary of effective TL and could be avoided. The conducted experiment showed that the opposite of the factors that cause effective TL are responsible and therefore, hypothesis 0, *“Semantically related topics have a similar probability distribution, and will have in TL no impact on PT or NT.”* is rejected and, the contrary hypothesis 1, *“Semantically related topics have a similar probability distribution. Therefore, chances for TL to lead a PT are likely”* is accepted.

5.2. Discussion

This research showed that TL is a promising technique that if used right can have massive advantages for businesses and researchers. It found a way how TL can be used effectively and how the NT can be avoided.

A full transfer (domain adaptation TL), trained only on the source and transfer to the target does not work well and, in this case, causes many NT. In some situations, this approach could be useful, for instance when there are no labels on the target, or when the target suffers from class noise (J. Pan et al., 2016). The combined-TL presented in this thesis is called in literature multiple source transfer learning (MSTL) and the power of this is confirmed in various studies (Ge et al., 2014; Huang, Wang, & Qin, 2012). The increase of training samples (within MSTL the source data is combined with 70% of target data) and optimized pre-model from the target dataset created by topic modelling, responsible for the good results (Hofmann, 2017).

On the algorithms can be seen that mostly NB works the best in this situation of TL. NB is a simple but often effective because of its probability-based modeling which is useful with unknown data (Lu et al., 2015). A side-effect is that NB works in the experiment as fastest algorithms.

The experiment done with deep learning with a neutral net caused fine results on ML-model but did not very well in TL. This is probably because the high layer neurons are specialized and optimized for the source task and this cause bad transfer rates (Yosinski, Clune, Bengio, & Lipson, 2014). Apparently deep learning is not the solution for everything, at least not for TL in this experiment.

TL is often not used by businesses because of the fear for NT (Rosenstein et al., 2005). The experiment conducted in this research shown that NT happens oft when datasets are not similar enough. In this research worked with review data and the TL between products-reviews works relatively well. Within products-reviews is seen that more similar products like Health & Personal care products and Beauty products-reviews even works better. This similarity effect is confirmed in multiple studies (Gui et al., 2017; Rosenstein et al., 2005; Torrey & Shavlik, 2009). The automotive dataset uses products that are further away from Beauty and HPC, and therefore they lead to more NT. The similarity between Amazon Automotive (car accessories) and Edwards (car reviews) is also larger as expected at forehand which results in a large number of NT. Therefore, to avoid NT the datasets should use as similar as possible in the distribution of words. Beauty is interesting because if this is used as a source it causes the most NT and used as target it works well. Probably due noise in the dataset (Gui et al., 2017) which creates bad transfers from when used as a source and in contrary, good results when used as target. HPC and Automotive have the lowest NT ratio.

The SMOTE sampling technique is considered as the “de facto” standard in imbalanced datasets (Fernández, García, Herrera, & Chawla, 2018) and gave the experiment also a performance boost on the minor classes neutral and negative. All four original selected datasets have the imbalanced problem. The class imbalance should be avoided to keep more clearance about the outcomes, therefore, are Amazon’s most balanced review dataset CPA is used to see how the TL works.

Interesting to see is that NB on Amazon CPA causes all NT and kNN works the best on this dataset. With this result can be concluded that a more balanced dataset has an effect on the algorithm/model and indirect on the success of TL. The sampling technique presented in this thesis improves the results, but also leads to overfitting. The experiments done showed that a sampling method that works for one dataset could perform worse on a different dataset (Ge et al., 2014). Probably the best solution is to increase the data volume like some experiments showed with 100.000 examples but that also increases the processing time factor 10. Latent Semantic Indexing gives the vector created by TF-IDF a dimension reduction and leads to a more accessible to compare vector with topics and leads to better transfer rates but also to a loss of information (Y. Zhu et al., 2011).

5.3. Practice recommendations

Before businesses invest in new technology, they need to know how they are going to profit from TL. This thesis showed a lot of profits that businesses and researchers could have from TL, but the benefits depend on the business case and the available data. This research have showed how TL can be used in practice, replace the amazon dataset with customer communication and it will show a starrating which can say something about the customer satisfaction.

5.4. Recommendations for further research

The hope is that this thesis could be a starting point for businesses and researchers to start working with TL. The work what is presented contains interesting results and conclusions but due to time issues its chosen to drop some ideas. In this research class balancing was a major issue and despite all effort in further work should this be even more important. The similarity of the word distribution is now inspected by with WordNet and by dataset metrics, but another way is to calculate the similarity with the KL-divergence (Kullback & Leibler, 1951). This was out of scope for this thesis, but can measure the distance between the word distribution (Moreno, Ho, & Vasconcelos, 2004). The extra experiments with deep learning & neural nets showed no promising TL results, but maybe there is a way that this can be optimized because also pre-trained models often work with deep learning and this needs further research.

5.5. Reflection

In this section is reflected on the quality of research and the validity of the conclusions. This research reviewed many papers used which increases the validity, but a guarantee that all important papers are found cannot be given. The experiment is performed with four datasets and is, due to time limitation ok, but when there was more time the number should be increased to realize a more unobstructed view. The sample size could also be increased to improve more reliability in minority classes. Some datasets/models are tested with 100.000 samples to guarantee the validity and reliability but showed only a small increase in performance but an exponential high processing time. To be fully sure it would be better to use all available data. The imbalanced datasets lead, despite of sampling techniques, to high accuracies due to the majority class researches high results. Therefore, will it be good to used balanced data to train from, if necessary synthetic created. Then should a class reduction not be necessary. Another option is to remove the neutral class and only use positive or negative.

The data is taken as authentic business data but how much of them contain fake reviews? Companies who receive reviews often ask their customers to give an excellent rating online, and in exchange, they get a discount the next time. This affects the business value and the quality of results. All used workflows and data is online available and can be reproduced.


6. References

- Anderson, E. W. (1998). Customer satisfaction and word of mouth. *Journal of service research*, 1(1), 5-17.
- Asghar, N. (2016). Yelp Dataset Challenge: Review Rating Prediction. *arXiv preprint arXiv:1605.05362*.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan), 993-1022.
- Boisot, M., & Canals, A. (2004). Data, information and knowledge: have we got it right? *Journal of Evolutionary Economics*, 14(1), 43-67.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide.
- Chatterjee, P. (2001). Online reviews: do consumers use them?
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
- Deacon, H. J., & Deacon, J. (1999). *Human beginnings in South Africa: uncovering the secrets of the Stone Age*: Rowman Altamira.
- Do, C. B., & Ng, A. Y. (2006). *Transfer learning for text classification*. Paper presented at the Advances in Neural Information Processing Systems.
- Egan, T. M., Yang, B., & Bartlett, K. R. (2004). The effects of organizational learning culture and job satisfaction on motivation to transfer learning and turnover intention. *Human resource development quarterly*, 15(3), 279-301.
- Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., & et al. (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22), 2199-2210. doi:10.1001/jama.2017.14585
- Engels, M. (2017). Data Analytics Task 2 Amazon & Yelp.
- Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of artificial intelligence research*, 61, 863-905.
- Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137-144.
- Ge, L., Gao, J., Ngo, H., Li, K., & Zhang, A. (2014). On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4), 254-271.
- Griffiths, O., Johnson, A. M., & Mitchell, C. J. (2011). Negative transfer in human associative learning. *Psychological science*, 22(9), 1198-1204.
- Gui, L., Xu, R., Lu, Q., Du, J., & Zhou, Y. (2017). Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, 1-13.
- Haskell, R. E. (2000). *Transfer of learning: Cognition and instruction*: Academic Press.
- He, R., & McAuley, J. (2016). *Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering*. Paper presented at the Proceedings of the 25th International Conference on World Wide Web.
- Helms, R. W. (2015). Datasafari – exploreren om te innoveren. In: Open University Press.
- Hofmann, T. (2017). *Probabilistic latent semantic indexing*. Paper presented at the ACM SIGIR Forum.
- Hu, N., Pavlou, P., & Zhang, J. (2006, 2006). *Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of Online word-of-mouth communication*.
- Hu, X., Pan, J., Li, P., Li, H., He, W., & Zhang, Y. (2016). Multi-bridge transfer learning. *Knowledge-Based Systems*, 97, 60-74.
- Huang, P., Wang, G., & Qin, S. (2012). Boosting for transfer learning from multiple data sources. *Pattern Recognition Letters*, 33(5), 568-579.
- Jurafsky, D. (2000). *Speech & language processing*: Pearson Education India.
- Kocaguneli, E., Menzies, T., & Mendes, E. (2015). Transfer learning in effort estimation. *Empirical Software Engineering*, 20(3), 813-843.

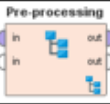





- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. Paper presented at the Advances in neural information processing systems.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79-86.
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3), 259-284.
- Leberman, S., McDonald, L., & Doyle, S. (2006). *The transfer of learning: Participants' perspectives of adult education and training*: Gower Publishing, Ltd.
- Lin, Y.-S., Jiang, J.-Y., & Lee, S.-J. (2014). A Similarity Measure for Text Classification and Clustering. *IEEE Transactions on knowledge and data engineering*, 26(7), 1575-1590. doi:10.1109/TKDE.2013.19
- Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems*, 80, 14-23.
- Makūnas, D. (2018). detectlanguage.com. Retrieved from www.detectlanguage.com
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.
- Moreno, P. J., Ho, P. P., & Vasconcelos, N. (2004). A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. Paper presented at the Advances in neural information processing systems.
- Muller, A. (2017). word_cloud Github. Retrieved from https://github.com/amueller/word_cloud
- Pan, J., Hu, X., Li, P., Li, H., He, W., Zhang, Y., & Lin, Y. (2016). Domain adaptation via Multi-Layer Transfer Learning. *Neurocomputing*, 190, 10-24.
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.
- Perkins, D. N., & Salomon, G. (1992). Transfer of learning. *International encyclopedia of education*, 2, 6452-6457.
- Perkins, D. N., & Salomon, G. (2012). Knowledge to go: A motivational and dispositional view of transfer. *Educational Psychologist*, 47(3), 248-258.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*: " O'Reilly Media, Inc."
- Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). *Self-taught learning: transfer learning from unlabeled data*. Paper presented at the Proceedings of the 24th international conference on Machine learning.
- Rosenstein, M. T., Marx, Z., Kaelbling, L. P., & Dietterich, T. G. (2005). *To transfer or not to transfer*. Paper presented at the NIPS 2005 Workshop on Transfer Learning.
- Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research methods for business students*. Harlow; Munich [u.a.]: Pearson.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), 1-47. doi:10.1145/505282.505283
- Torrey, L., & Shavlik, J. (2009). Transfer learning. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1, 242.
- Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data - Journal Article*, 3(1), 1-40. doi:10.1186/s40537-016-0043-6
- Woodworth, R. S., & Thorndike, E. (1901). The influence of improvement in one mental function upon the efficiency of other functions.(I). *Psychological review*, 8(3), 247.
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks?* Paper presented at the Advances in neural information processing systems.
- Yu, D., & Deng, L. (2014). *Automatic speech recognition: A deep learning approach*: Springer.
- Zhu, X., & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3), 177-210.

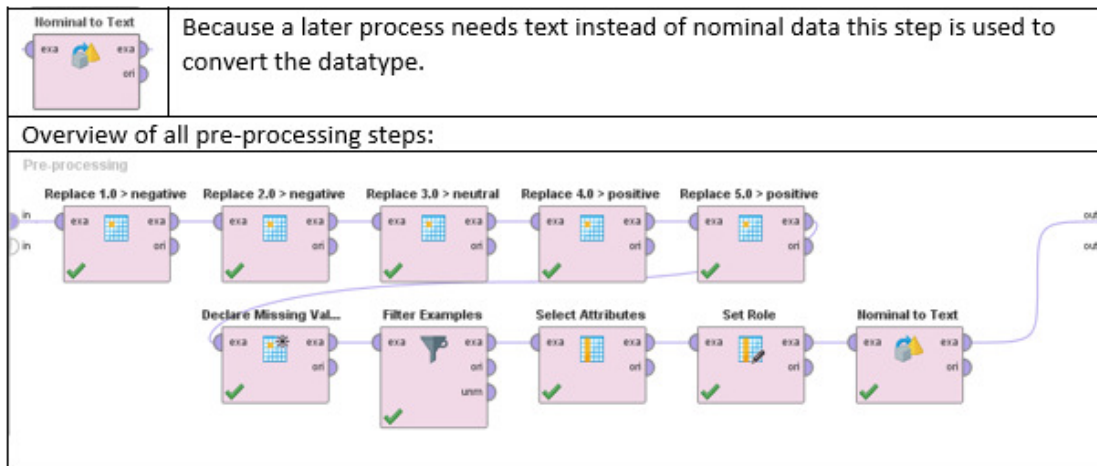
Appendix 1; Explanation of all used operators

Step 1 - Read CSV

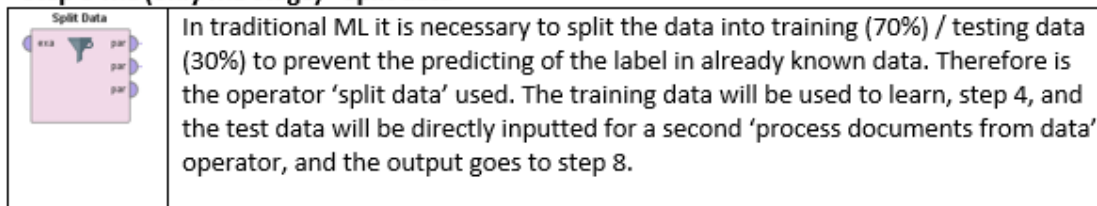
	<p>This operator is responsible for reading the already prepared CSV file (missing values, outliers, other languages reviews are already removed), and has the advantage that the following steps can be short and improves the performance of the model.</p> <p>The official name is renamed to the dataset that is getting imported to keep clear which data is processed. In the TL-stage a second read CSV operator is added to import the target dataset.</p>
---	--

Step 2 - Pre-process steps

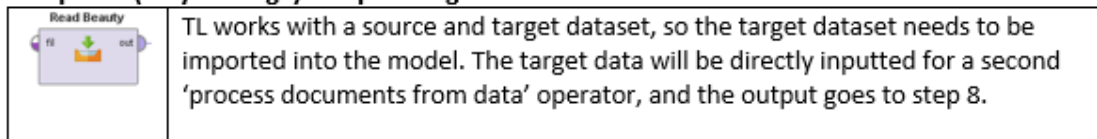
	<p>After importing the data, the subprocess 'pre-process' is used to pre-process data the further steps will explain the content. The declare missing values, and filter examples are added to improve the generalization of the model.</p>
	<p>The datasets are massively unbalanced, and people interpret text and star rating sometimes different so that words and star rating not always match. To reduce the effect and make the predictions clearer the 5-star rating is reduced to a 3-star rating with negative, neutral or positive values to increase the accuracy of predictions.</p> <p>This could also be done with the operator 'discretize by binning' but requires a different import format which encounters with later operators, so it makes more sense to use this natural and more transparent, to repeat the operator for every class.</p> <p>Corrected values:</p> <ul style="list-style-type: none"> 1.0 > negative 2.0 > negative 3.0 > neutral 4.0 > positive 5.0 > positive
	<p>There are no missing values in the dataset anymore but to keep the model portable for future datasets the 'declare missing values' operator is added and marks missing values</p>
	<p>This operator filters out missing values</p>
	<p>The confrontation matrix has shown that the relation between the other attributes is low and we focus on learning on text and no other features except the 'overall', and 'reviewText' are filtered out.</p>
	<p>The operator 'set role' sets the label to the 'overall' attribute so that the classifier know that this value needs to be predicted.</p>



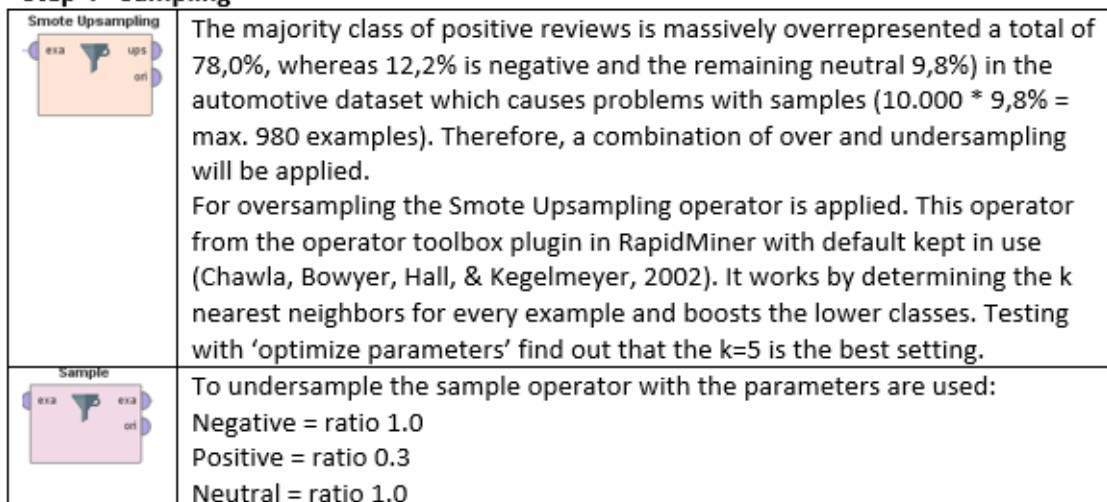
Step 3-ML (only ML-stage) - Split data









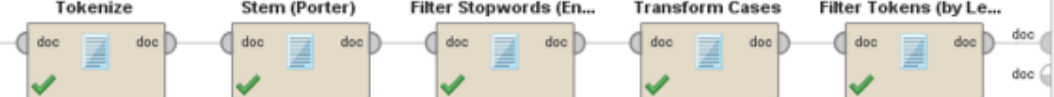
Step 3-TL (only TL-stage) – Import target data




Step 4 - Sampling



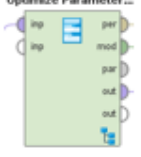
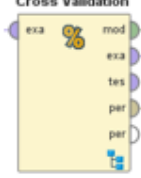




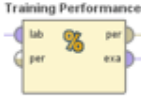
Step 5 - Process documents from data

	<p>To transform unstructured data into structured data, the operator “Process Documents from Data” is used. This operator is available in RapidMiner after installing the extension “Text Processing” and is also a subprocess.</p> <p>A simplistic approach in text mining is Bag of Words. This approach treats every document as a collection of individual words (tokens) from the corpus and is useful but ignores grammar and importance of words. A second approach is Term Frequency (TF) this counts how often a word occurs in a document, the more often this appears in the document, the more important it is.</p> <p>For instance when the adjective word “great” is found in more documents then this is probably more important than the word “adapt” which occur only a few times in the corpus. Even more advanced is TF/IDF, based on the multiplication of TF and inverse document frequency (IDF) and indicates how characteristic a term is for the corpus. Words should not too rare but not too familiar either (Engels, 2017).</p> <p>Therefore is TF/IDF used to count how often a word appears and in how many documents it appears. TF/IDF uses this information to calculate the importance of the words by assigning them weights. This creates a vector for each word (Ramos, 2003). Also is tried to use the Wordnet stemmer, find hypernyms, hyponyms, and synonyms but this did not improve the accuracy.</p>
	<p>Tokenize (non-letters, transform every document of the corpus into single words, tokens).</p>
	<p>Stem (by the Porter algorithm and by calculating syllables to reduce the number of words with the same meaning and find the lemma, for example, “stemmer”, “stemming” and “stemmed”). Also are other stemmer tried like the WordNet stemmer but this resulted in lower accuracy after classifying.</p>
	<p>Filter Stopwords (filters out the English words like do, the, from, etc.)</p>
	<p>Transform Cases (normalizing, change to lowercase, “Hello” and “hello” are the same</p>
	<p>The operator ‘filter tokens by length’ removes words longer then 15 characters and less than two characters.</p>
<p>Overview of all “Process documents from Data” operators</p>	
	




Step 6 – Latent Semantic Indexing with Topic Modeling and Dimension Reduction

	<p>The TF/IDF results in thousands of features. To perform dimensionality reduction the SVD operator (Singular Value Decomposition) is used. This RapidMiner operator uses ‘latent semantic analysis’ which is a method for finding topic models at large corpora of text, and convert the vector to topic models which represent the same topics (Landauer et al., 1998).</p> <p>Optimizing parameter operator showed that 100 dimensions is the optimal parameter.</p>
---	--

Step 7 – Optimize Parameters, Cross-validation

	<p>Optimizing parameter operator is a subprocess to find the optimal parameter settings. Therefore, is during the developing phase every operator or process placed in there to find the optimum.</p> <p>The optimum for classifying depends on the dataset therefore is the complete cross-validation and are all results run in this operator. This could lead to less PT's and more NT's and severe increase of processing time but better results.</p>
	<p>To classify the data will be tested several algorithms: kNN, Naïve Bayes, and GLM. The first two are selected because they are common algorithms and GLM because this gives excellent results in the 'automodel' function of RapidMiner. The algorithm counts the negative and positive words in a review and predicts to which class a review belongs. This can be difficult, because words often have a different degree, like "good" and "great". This causes words to be classified in a higher or lower class; the algorithm should learn this by giving weights to the words. Also, words often combined. For example, "not good" and "good" where the combination of words is essential and context based (Provost & Fawcett, 2013).</p>
	<p>k-Nearest Neighbor is a classifier algorithm, placed inside training of cross-validation, that tries to find similar cases learned from the past (ML: training/TL: source) and helps to predict cases in the future (ML: testing/TL: target). Where the number of k is the number of closest neighbors. To find the optimal value for k the optimize parameters operator is used.</p>
	<p>Naïve Bayes is a simple to use an algorithm that counts the number of occurrences of an example about the whole dataset. It has not many parameters to change so only the laplace_correction (boolean) will be tested. This parameter is used to support the weakness of Naïve Bayes by not counting the zero values and present better results.</p>
	<p>Generalized Linear Model is the RapidMiner name for the H2O algorithm and give good results on single datasets and will be used to test in TL.</p>
	<p>The operator 'apply model' is inside the testing part of cross-validation and is used to present the learning algorithm unseen data.</p>
	<p>The performance operator, here renamed to 'training performance' to get easier recognized in the results. Selected are: main criterion: accuracy, classification error, weighted mean recall, mean precision, logistic loss and skip undefined labels.</p>

Step 8 – Testing/transferring

	<p>Again, is the apply model operator used this time to get the new data (testing or transferred data from source) with pre-processing model parameters from SVD.</p>
	<p>Then combine it with the classifying model and make the predictions.</p>
	<p>Do the performance evaluation and show the results to the output. Selected are: main criterion: accuracy, classification error, weighted mean recall, mean precision, logistic loss and skip undefined labels.</p>