NTNU

Norwegian University of
Science and Technology

Automatic Classification of Bank Transactions

Olav Eirik Ek Folkestad
Erlend Emil Nøtsund Vollset

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Jon Atle Gulla, IDI

Norwegian University of Science and Technology
Department of Computer Science

Preface

This is an in-depth study of different machine-learning approaches used in the classification of bank transactions. The study has been carried out at the Norwegian University of Science and Technology during the spring of 2017. The project has been carried out in collaboration with Sparebank1 SMN.

Trondheim, 20-06-2017

Erlend Vollset

Eirik Folkestad

Acknowledgment

We would like to thank the following persons for their great help during the development of this thesis.

First and foremost, we would like to thank our supervisor Jon Atle Gulla for providing us with counsel and feedback throughout the entire process.

Furthermore we would like to express our gratitude to the team at Sparebank1 - specifically Marius Rise Gallala, Ragnar Furunes, and Stian Fagerli Arntsen - who have provided us with our data and spent their time answering our questions and giving us valuable feedback.

We also want to thank Iver Jordal for excellent advice on methodology and different approaches to the problem.

E.V. & E.F.

Contents

Chapter 1

Introduction

This first chapter contains an overview of the problem at hand and presents which research areas we have chosen to focus on. The chapter is concluded by a structure outline of the entire project.

1.1 Background

Sparebank1 currently uses a manual filter to classify transactions. This requires a good deal of maintenance to keep updated and there are several categories being omitted from classification because the filter does not pick them up. They would therefore like to develop a system which can improve and automate this procedure.

Ultimately, this system is intended to be used as a foundation for analyses internally in the bank in order to identify consumption trends. It can also be used to provide their customers with a thorough and accurate overview of their finances.

1.2 Problem Formulation

This is a multi-class classification problem and our goal is to investigate which supervised machine learning methods are best suited to solve it. We are also using external semantic resources to supplement the information we have about each transaction and in turn attempt to increase the accuracy of the classification system. Determining which external semantic resources we

should use and how this is to be implemented is therefore an important aspect of this project.

1.3 Research Questions

The main objectives of this project are to answer the following research questions:

1. *How do Logistic Regression and Feed-Forward Neural Networks compare in the classification of bank transactions?*
   We wish to perform a thorough analysis of the performance of these two approaches. The experiments we conduct will look at both the accuracy and the response time of the different approaches.

2. *Can external semantic resources like Brønnøysundregisteret and Google Places be used to improve the accuracy of the classification system?*
   We wish to explore how we can use external semantic resources to improve the classification system. In addition to finding the best methods for exploiting this data, we would like to measure to what degree they improve the accuracy of the system.

3. *Can linked open data sources like DBPedia and WikiData be used to improve the accuracy of the classification system?*
   For this final research question, we wish to explore how we can use linked data to provide our data set with more information about each transaction and in turn hopefully improve the accuracy of the classification system. We will evaluate different linked open data sources and provide an analysis of these results.

1.4 Results and Conclusions

The main results of the approaches we have implemented are presented in Figure 1.1 as the difference in evaluation scores between the approach and the baseline. Here we can see that the semantic resources *Brønnøysund Registry* and *Google Places API* improve the accuracy of the classification system, while *Wikidata* and *DBpedia* lead to a decrease in accuracy.

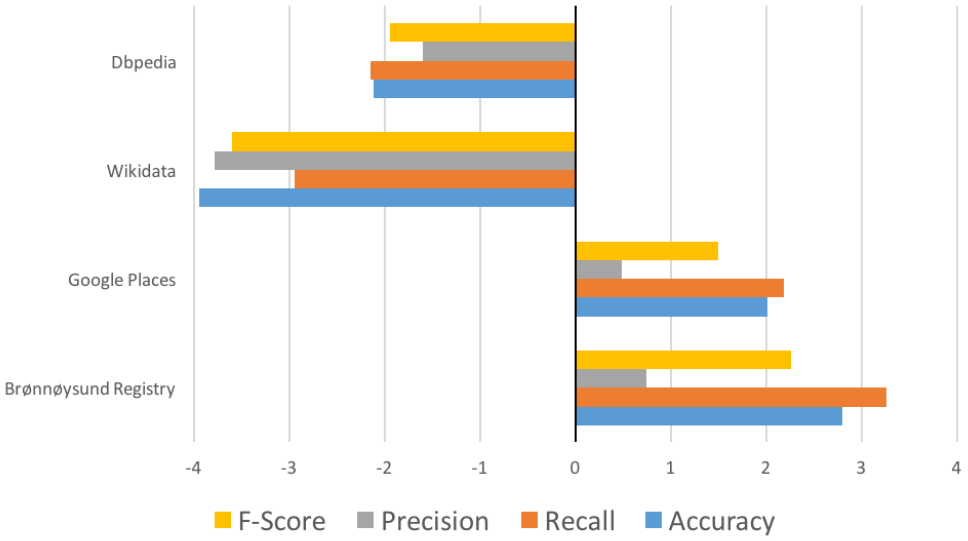

Figure 1.1: Comparison of approach evaluation score improvements. Measured in percentage points.

Our final system, therefore, builds on a standard machine learning approach and a feature enrichment approach using the *Brønnøysund Registry* and the *Google Places* API as external semantic resources, ultimately leading to the evaluation scores shown in Table 1.1.

Accuracy	**Recall**	**Precision**	**F-Score**
0,9464	0,9320	0,9554	0,9430

Table 1.1: Results for final approach

1.5 Limitations

This section describes factors which result in a reduction of the scope of the project.

Data limitations

Given the confidential nature of the data we are dealing with, certain parts of the data set must be anonymized or removed before we are permitted to work with it. This anonymization results in a more generic data set with instances which are harder to classify and therefore reduces the expected overall accuracy of the system. The training and test data we use has been labeled manually, which means that it may be subject to errors. However, we believe that this error is negligible and will not affect the classification results to a significant magnitude.

Computational limitations

The system we have produced is not intended to be used as a large-scale, real-time application, but as an evaluation environment for different classification approaches specific to this problem. Computational efficiency and response time are therefore not evaluated in our approaches but may be subject to discussion.

1.6 Structure of the Report

The rest of the report is structured as follows. Chapter 2 describes the theoretical foundation upon which we have built our project. It also provides an overview of the testing environment we have developed. Chapter 3 follows with a presentation of the data we have used and explains, in detail, how we have preprocessed it. Chapter 4 will continue with a thorough explanation of the external semantic resources used. We will cover how this data is extracted and how it is processed. Chapter 5 will provide a summary of the research papers we have written in conjunction with this thesis. These articles will take an in-depth look at the methods we have used in our classification system. Chapter 6 presents what experiments we have conducted, as well as the results we obtained from them. The project is summarized in Chapters 7 and 8 by discussing our findings, providing recommendations for further work, and drawing our final conclusions in relation to the research questions.

Chapter 2

Background

This chapter lays the theoretical foundation of the project. Here we explain the main concepts and techniques we have applied in our approach.

2.1 Classification

In machine learning, classification is considered an instance of supervised learning[6]. Given a set of categories, it is the process of identifying to which of these categories a new observation belongs. This is done based on a training data set containing observations and their corresponding target categories.

Some common real-world classification problems are diagnosing patients based on their symptoms and medical history, determining whether or not e-mails are spam based on their content and sender, or classifying images of hand-written characters in order for machines to read hand-written texts. There is a vast landscape of application areas for classification techniques, and an even more challenges to accompany them.

2.2 Supervised vs. Unsupervised Learning

Supervised learning is based on training a model using a *labeled* data set[16]. This model is then used to map new observations. Unsupervised learning is a different approach where patterns are identified and a function is inferred from a set of *unlabeled* data.

The unsupervised counterpart of classification is known as clustering. This technique does not bind an observation to a label, but simply groups it together with the previous observations which are most similar.

2.3 Structured vs. Unstructured data

Structured data refers to information with a high degree of organization. This means that it can be easily transferred to a relational database and facilitates querying and searching. An example of this is the phone book, here we have structured information with a given feature set. The data set used in this project is another example, which is described in Chapter 4.

Unstructured data refers to data which is not organized in any specific manner, making it difficult for computers to extract meaning from them. The most common form of unstructured data is free-form text such as in news articles or e-mails.

2.4 Feature set

A feature set, or feature-vector, is a vector of n numerical features intended to describe some object. This representation facilitates processing and statistical analysis, which is a necessity in most machine learning algorithms.

2.5 Baseline System

A baseline refers to a set of techniques and configurations applied to our system intended to serve as a basis for defining change and measuring improvement. Our baseline system is a standard machine learning approach to text classification which involves using a Bag-of-Words representation and Logistic Regression. We have chosen to use this model because we believe our data to be linearly separable. Also, linear models are robust and tend to need much less handholding than more complex approaches [17].

2.6 Bag-of-Words Model

The Bag-of-Words Model is used to convert the transaction descriptions to a representation better suited for machine learning. This particular technique is commonly used in natural language processing and information retrieval. In our application of the model, it is used as a tool for feature generation. When generating features for a corpus of texts, each text is represented as a multiset (bag) of the terms contained in the text. Given a corpus of texts $X = x_1, x_2$, where

x_1 = Alan has a chair

x_2 = A chair is a chair

the bag-of-words representation produced is shown in Figure 2.1.a. The resulting matrix has a column for each term in the corpus and a row for each text. The value is the term frequency, i.e., the number of occurrences of the term in a given text. These features may then be used as input to a predictive model such as the one in this project.

X	Alan	has	a	chair	is
x1	1	1	1	1	0
x2	0	0	2	2	1

(a) Bag-of-Words

C1	1	0	0
C2	0	1	0
C3	0	0	1

(b) One-Hot Encoding

Figure 2.1: Representation Examples

2.7 One-Hot Encoding

One-Hot is a sequence of bits where a single bit is 1, and the rest are 0. One-Hot Encoding is a method for representing a set of features using One-Hot bit sequences. The length of the sequence of bits is equal to the size of the set of features. The bit which represents the given feature is 1 and all others 0. Assume three categories denoted as C_1, C_2, and C_3, their One-Hot encoded representation is shown in Figure 2.1.b.

The feature being represented is projected onto a plane, and all the produced planes are at an equal distance from each other. This categorical representation ensures that there is no

ordinal relationship between the features. This makes it ideal for representing non-numerical features. We have used this technique to represent certain external data elements.

2.8 Logistic Regression

The Logistic regression classification algorithm is linear and estimates a probability of a class Y given a feature-vector X. It does this by using a logistic function to find the relationship between the class and the feature-vector. It assumes that the distribution P(Y|X), where Y is the class and X is the feature-vector, is on a parametric form and then estimates it from the training data. The probability P(Y|X) of X belonging to class Y is given by the sigmoidal function which we can see in Eq. 2.1 and Eq. 2.2.

$$z(Y,X) = \sum_{i=1}^{N} w_i f_i(Y,X) \tag{2.1}$$

$$P(Y|X) = \frac{1}{1+exp(-z(Y,X))} \tag{2.2}$$

P(Y|X) is estimated by linearly combining the features of X multiplied by some weight w_i and applying a function $f_i(Y,X)$ on the combinations. f_i is a function which returns a relationship value between a feature of a class and a feature in a feature-vector in the form of true or false based on the probability being over a certain threshold. Some features are more important than others, so the weight w_i denotes the "strength" of the feature.

The Logistic Regression classifier uses a discriminative algorithm which means that it can compute $P(A|B)$ directly, without the need to compute the likelihood of $P(B|A)$ first. From Logistic Regression's discriminative properties it can be assumed that it has relatively low asymptotic error compared to the generative approach but will require a larger set of training data to achieve this.

A multi-class example of Logistic Regression is the One-vs-Rest approach. Here a classifier is trained for each class. These classifiers predict whether or not an observation belongs to the class. Then, to classify new observations, you pick the class whose classifier maximizes the probability of the observation belonging to it. In figures (a), (b), and (c) in Figure 2.2, data from

each individual class has been fit to their respective classifiers.

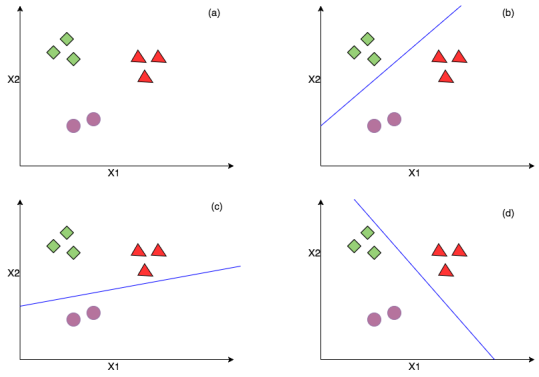

Figure 2.2: Logistic Regression example
(a) feature-vectors | (b) classifier for diamonds
(c) classifier for circles | (d) classifier for triangles

Another multi-class example of Logistic Regression is to replace the logistic function in Eq. 2.2 with the SoftMax function as we see in Eq. 2.1 and Eq. 2.3 [7]. An example of the Softmax classifier can be seen in Fig. 2.3.

$$P(y|X)_{y\in Y} = \frac{exp(z(y,X))}{\sum_{y'\in Y} exp(z(y',X))} \tag{2.3}$$

From the expression in 2.3 it can be shown that $\sum_{y\in Y} P(y|X) = 1$, which leads to a classifier for a feature-vector X which outputs the class $\hat{y}$ if only the class of the feature-vector is needed and not the probability itself.

$$\hat{y} = argmax_{y\in Y} P(y|X) \tag{2.4}$$

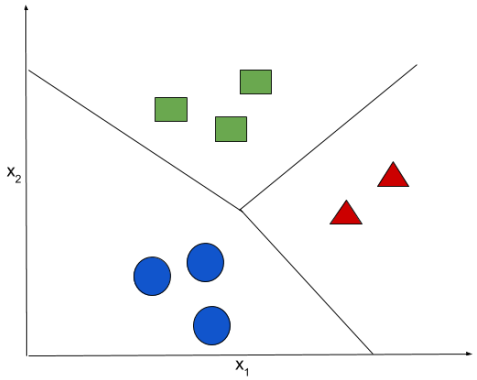

Figure 2.3: Softmax Regression example

2.9 Feed-Forward Neural Network

To examine whether or not the problem is truly linearly separable, it may be a sensible approach to implementing a classifier which can handle non-linear classification. A feed-forward neural network with one or more hidden layers can classify data when a more complex classification function is needed [7]. When used as a classifier the FFNN learns a function approximator that maps a feature-vector X to a class Y after having trained on a dataset. An FFNN uses one or more neuron layers, called hidden layers, in addition to the input layer and output layer to learn the approximation of a function that can be used to classify the input. The first neuron layer represents the feature-vector denoted here as (see the orange node in 2.4).

$$\{x_i | x_1, x_2, ..., x_m\} \tag{2.5}$$

The input layer takes in the feature-vector in a suitable format (see green nodes in 2.4). Each subsequent hidden layer (see blue nodes in 2.4) will transform the data for each neuron in the layer with an integration function followed by an activation function before outputting the data to the next layer. The integration function calculates a weighted linear summation of the neurons in the previous layer (see Eq. 2.6). The weight denotes the importance of a neuron.

$$w_1 x_1 + w_2 x_2 + ... + w_m x_m = s_1 + s_2 + ... + s_m \tag{2.6}$$

The activation function maps the output of the integration function to a normalized fixed domain which represents throughput of the neuron or in other words how much this neuron contribute to the next layer. The activation function is often a logistic non-linear function able to handle non-linear classification problems. However, it is shown that approximations with activation functions like Rectified Linear Unit can perform just as well for non-saturating non-linearity problems [20].

$$net_i \leftarrow w_{i0} + \sum_{j \in Pred(i)} (w_{ij} x_j) \tag{2.7}$$

$$a_i \leftarrow f_{sig}(net_i) \tag{2.8}$$

The output layer transforms the results of the last hidden layer to values that can be output like classes (see the red nodes in 2.4). FFNN is most commonly trained using the backpropagation algorithm where the approximation error is propagated backward in the network, and the weights in the network are updated by an optimization algorithm inspired by gradient descent. There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following [13]:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

FFNN is especially useful when classes are of patterns that are not linearly separable, and a complex function is needed to estimate classes. Artificial Neural Networks, like FFNNs, are also effective on data of high dimensionality. However, a large data set is needed to train the neural network to perform well. Furthermore, the results are difficult to justify as the resulting function may be hard to inspect due to the hidden layers [17].

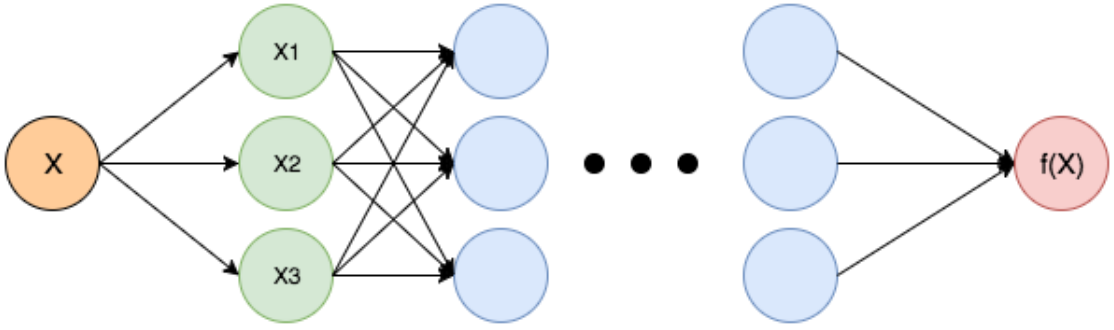

Figure 2.4: Feed-Forward Neural Network

2.10 Linked Open Data

Linked Open Data is about connecting structured, distributed data across the web. This machine-readable network of data is known as the semantic web and uses a variety of technologies to provide an environment where applications can publish and query data, reason on knowledge, and draw inferences using vocabularies[23]. These technologies include, among others:

- **RDF**, which is the underlying data model for the semantic web consisting of a Subject, a Predicate, and an Object [11].
- **Ontologies**, which define the concepts and relationships (also referred to as “Terms”) used to describe and represent an area of concern. The trend is, however, to use "ontologies" about more complex, and possibly quite formal collection of terms, whereas “vocabulary” is used when such strict formalism is not necessarily used or only in a very loose sense. [2].
- **SPARQL**, which is the query language used to express queries across the diverse data sources on the semantic web[19].

There are many linked open data sets on the web which are all accessible by anyone at any time. Among these are DBpedia and Wikidata, which are important benefactors in the linked data community.

The applications for linked open data are many. In data mining it has been applied to support of complex and inter-disciplinary data mining analysis [15]. In big data the value of discovered knowledge could be of greater value if it is available for later consumption and reusing. Sharing results as linked open data can benefit in many other applications [9]. Linked open data has also been used with success in Query Expansion which is the the process of reformulating queries to improve retrieval performance in information retrieval operations [24].

2.11 Evaluation Metrics

Evaluation metrics are used to assess classification models. In this section, we will explain which metrics are required to perform a robust evaluation of multi-class classification models. It is

important to assess the performance in each class to evaluate the inner workings of the multi-class classification model. To explain how the metrics we have used work, we will introduce the following basic measures on the example class A:

- **True Positive** (TP) refers to a data instance being correctly classified. Ex: Instance of class A classified as A.
- **False Positive** (FP) refers to a data instance being falsely classified to a given class. Ex: Instance of class B classified as A.
- **False Negative** (FN) refers to a data instanced being falsely rejected from a given class. Ex: Instance of class A not classified as A.
- **True Negative** (TN) refers to a correct rejection of classification. Ex: Instance of class B not classified as A.

The relationship between these basic performance measures is displayed in Table 2.1.

	True Label A	True not A
Predicted Label A	true positive (**TP**)	false positive (**FP**)
Predicted not A	false negative (**FN**)	true negative (**TN**)

Table 2.1: Performance table for instances labeled with class A[1]

Accuracy

Accuracy is the number of correct predictions made to the total number of data instances. It is expressed as

$$\frac{\sum_c^C TP_c}{N} \tag{2.9}$$

Where N is the total number of data instances, C is the number of different classes, and TP_c is the number of true positives for class c. This is the same as micro-averaged recall (explained below) and will be referred to as this in the remainder of the report.

[1]http://www.cnts.ua.ac.be/ vincent/pdf/microaverage.pdf

In the context of multi-class classification, this metric can be misguiding. A weak model may achieve high accuracy for a biased data set. For example, if a given data set consists 90% of instances of class A and it classifies all instances to this class, it will achieve an accuracy score of 90%. This is misleading as the model falsely classifies all instances which do not belong to class A.

Recall

Recall is the number of correctly classified instances of a class to the number of instances of that particular class. It is expressed as

$$\frac{TP}{TP+FN} \tag{2.10}$$

and is visualized in Figure 2.5.

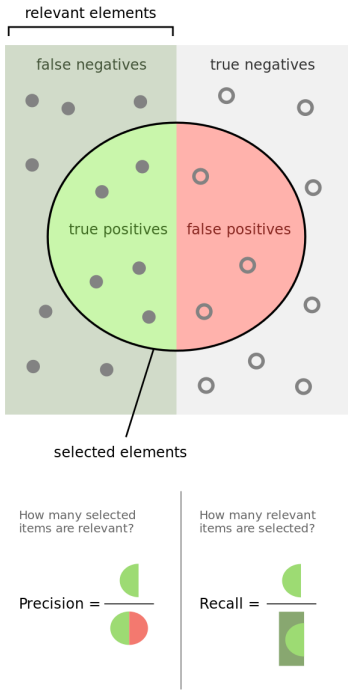

Figure 2.5: Precision and Recall visualized[2]

[2]https://en.wikipedia.org/wiki/File:Precisionrecall.svg

Precision

Precision is the number of correctly classified instances of a class to the number of instances classified as this class, falsely or not. It is expressed as

$$\frac{TP}{TP+FP} \tag{2.11}$$

and is visualized in Figure 2.5.

F-Score

The F-Score can be interpreted as a weighted average of the precision and recall, where it reaches its best value at 1 and worst at 0. We employ the balanced F-score (F1 score), which is the harmonic mean of precision and recall. It is expressed as

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{2.12}$$

Micro- and Macro-averaged scores

Recall and precision are good ways of measuring the model's performance with respect to individual classes. In order to boil all those class-specific numbers down to a metric which gives more general results, we introduce micro- and macro-averaged scores.

Macro-averaging normalizes the sum of precision/recall for each class using the total number different classes, so it does not consider the label distribution in the dataset. Macro-averaging is expressed as[22]

$$B_{macro} = \frac{1}{q} \sum_{\lambda=1}^{q} B(TP_\lambda, FP_\lambda, FN_\lambda, TN_\lambda) \tag{2.13}$$

where B is a binary performance measure (precision or recall) and $L = [\lambda_j : j = 1...q]$ is the set of all labels.

Micro-averaging computes the average using the sums of TP, FP, FN, and TN for all classes. This method, therefore, takes the frequency of the label in the dataset into consideration. This

average is expressed as[22]

$$B_{micro} = B\left(\sum_{\lambda=1}^{q} TP_\lambda, \sum_{\lambda=1}^{q} FP_\lambda, \sum_{\lambda=1}^{q} FN_\lambda, \sum_{\lambda=1}^{q} TN_\lambda\right) \tag{2.14}$$

2.12 Overview of Testing Environment

Here we present an overview of the testing environment we have built. We have used Scikit-Learn and TensorFlow, which are free, open source machine learning libraries for Python. The modules which constitute the testing suite are:

- **Classifier Module** which provides a framework for setting up classifiers from Scikit-Learn. It also contains the Feed-Forward Neural Network we built using TensorFlow.
- **Data Processing Module** which facilitates I/O tasks and preparing the data set for effortless handling in other modules.
- **Classification Module** which provides a framework for preprocessing data, fitting data to classifier models, and performing predictions.
- **External Resource Module** which provides methods and data structures for extracting and representing data from the different external semantic resources used in this project.
- **Evaluation Module** which provides a framework for evaluating the performance of classifiers and producing visualizations of these results.

Chapter 3

Related Work

In this chapter we present a few studies which are closely related to the work we are conducting in this project.

A project conducted by Skeppe[21] attempts to improve on an already automatic process of classification of transactions using machine learning. No significant improvements were made using fusion of transaction information in either early or late fusion. The results do however show that bank transactions are well suited for machine learning, and that linear supervised approaches can yield acceptable scores.

In Gutiérrez et al.[12] they use an external semantic resource to supplement sentences designated for sentiment classification. The resource and methods they propose reach the level of state-of-the-art approaches. In the study conducted by Albitar[5], classification of text is performed using a Bag-of-Words Model which is conceptualized and turned into a Bag-of-Concepts Model. This model is then enriched using related concepts extracted from external semantic resources. Two semantic enrichment strategies are employed, the first one is based on semantic kernel method while the second one is based on enriching vectors method. Only the second strategy reported better results than those obtained without enrichment.

Iftene et al.[14] present a system designed to perform diversification in an image retrieval system, using semantic resources like YAGO, Wikipedia, and WordNet, in order to increase hit rates and relevance when matching text searches to image tags. Their results show an improvement in terms of relevance when there is more than one concept in the same query.

In the research conducted by Ye et al.[25] a novel feature space enriching (FSE) technique to

address the problem of sparse and noisy feature space in email classification. The (FSE) technique employs two semantic knowledge bases to enrich the original sparse feature space. Experiments on an enterprise email dataset have shown that the FSE technique is effective for improving the email classification performance.

Poyraz et al.[18] perform an empirical analysis the effect of using Turkish Wikipedia (Vikipedi) as a semantic resource in the classification of Turkish documents. Their results demonstrate that the performance of classification algorithms can be improved by exploiting Vikipedi concepts. Additionally, they show that Vikipedi concepts have surprisingly large coverage in their datasets which mostly consist of Turkish newspaper articles.

Xiong et al.[24] present a simple and effective method of using a knowledge base, Freebase, to improve query expansion, a classic and widely studied information retrieval task. By using a supervised model to combine information derived from Freebase descriptions and categories to select terms that are useful for query expansion. Experiments done on the ClueWeb09 dataset with TREC Web Track queries demonstrate that these methods are almost 30% more successful than strong, state-of-the-art query expansion algorithms. Some of these methods also have 50% fewer queries damaged which yield better win/loss ratios than baseline algorithms.

In our research we have combined feature enrichment using external semantic resources with classification of real bank transactions. This is an important intersection that needs further research. We hope to have laid a foundation upon which others can continue research in the domain of classification of financial data.

Chapter 4

Data

This section describes our data source, how information about each transaction is represented, and how we have preprocessed the data.

4.1 Data source

Sparebank1 collects and stores information about all transactions which flow through their system. The bank transaction data set consists of 220618 unstructured Norwegian transaction descriptions which have been manually labeled by a third party company. These are actual bank transactions from a given time interval provided to us by Sparebank1 SMN, the central Norway branch of Sparebank1. SpareBank1 is a Norwegian alliance and brand name for a group of savings banks. The alliance is organized through the holding company SpareBank1 Gruppen AS that is owned by the participating banks. In total the alliance is Norway's second largest bank and the central Norway branch is the largest bank in its region.

- ***TRANSAKSJONSTEKST*** - An unstructured text string describing the transaction.
- ***UNDERKATEGORI_ID*** - An id number representing a sub-category of the main categories.
- ***UNDERKATEGORI_NAVN*** - A text string stating the name of the sub-category.
- ***KATEGORI_ID*** - An id number representing a main category.
- ***KATEGORI_NAVN*** - A text string stating the name of the main category.

- ***KATEGORISERT_VIA_KJEDE*** - A simple character 'J' or 'N' representing whether or not the categorization is done using KJEDE_NAVN.

- ***KJEDE_NAVN*** - A text string stating the name of the business associated with the transaction. This field is empty if no business is explicitly contained within TRANSAKSJONSTEKST. KATEGORISERT_VIA_KJEDE is then 'N.'

Features pertaining to time and account information have been removed as this data is sensitive. We do not have any knowledge as to how the third party company has labeled the transactions in this dataset other than having used a set of manually defined rules.

The data set is assumed to representative of the entire set of transactions flowing through Sparebank1s systems, and the manual labelings are assumed to be correct. This makes this data set ideal for training and testing an automated approach to our problem.

4.2 Data Representation

Transaction

Our data is represented as comma-separated-values, and a transaction may look like the one in table 4.1. The information about this transaction states that the transaction belongs to the main category with KATEGORI_ID 44, and the sub-category with UNDERKATEGORI_ID 62. The categorization has been done by finding the store or business since KATEGORISERT_VIA_KJEDE is 'J' and KJEDE_NAVN is 'Deli De Luca.' Our model only uses the transaction description as input and category ids as outputs; all other features will, therefore, be disregarded.

TRANSAKSJONSTEKST	UNDERKATEGORI_ID	UNDERKATEGORI_NAVN	KATEGORI_ID	KATEGORI_NAVN	KATEGORISERT_VIA_KJEDE	KJEDE_NAVN
DELI DE LUCA TORGGT. 8 OSLO	62	Kioskvarer	44	Dagligvarer	J	Deli De Luca

Table 4.1: Example Transaction

Categories & Sub-categories

Categories are represented using a name and an ID. A list of the possible main categories is shown in Table 4.2. All possible main categories and their sub-categories can be found in the appendix (See 11.1).

Main Category ID	Main Category Name	Main Category Name English
42	Bil og transport	Automobile and Transport
43	Bolig og eiendom	Housing and Real-Estate
44	Dagligvarer	Groceries
45	Opplevelse og fritid	Recreation and Leisure
47	Helse og velvære	Health and Well Being
48	Hobby og kunnskap	Hobby and Knowledge
49	Klær og utstyr	Clothes and Equipment
103	Annet	Other
104	Kontanter og kredittkort	Cash and Credit
181	Finansielle tjenester	Financial Services

Table 4.2: Categories and their IDs

4.3 Preprocessing of Data

Preprocessing of Original Data Set

As explained in section 4.1, Sparebank1 have provided us with a dataset which is considered to be a representative subset of their data. Certain features have been removed, and others have been edited to ensure the privacy of their users.

They use an algorithm to clean the textual transaction description. This cleaning procedure removes parts of the string which may contain sensitive information (e.g. contract number) or is described in other features (e.g. date, transaction amount). An example of this is shown in figure 4.1.

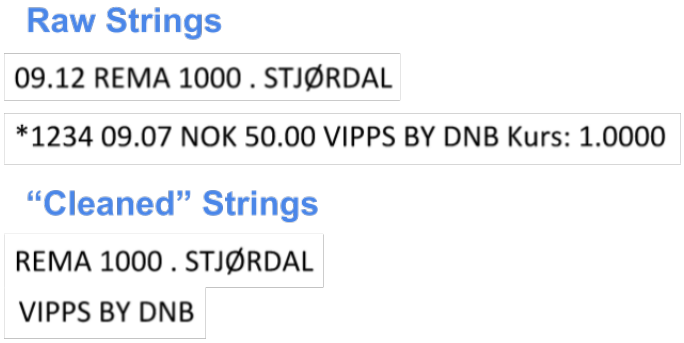

Figure 4.1: Example of stripping transaction information

Preprocessing of Data Set

As mentioned in section 4.2 we only use the textual description from the data received from Sparebank1 as input and category ids as target values. The other features are redundant in our implementation. In order to prepare these features for the algorithms we are using, we have performed the following preprocessing steps:

- **Clean Transaction Description** - Before the transaction texts can be used we preprocess them using regex to remove all non-alphabetical symbols. We also remove all words with less than three characters.

- **Bag-of-Words** - In order to work with the transaction texts, we need to extract feature vectors suitable for machine learning. That is, we need a numerical representation of the text to perform calculations. We have chosen to use the Bag-of-Words representation. Other methods, such as Word-2-Vec, are more sophisticated but are not required in this case. This is because we do not need any information about the token context as the transaction texts are only strings of keywords. This is also used to represent the data we receive as output from WikiData, DBpedia, and the Thesaurus.

- **One-Hot Encoding** - Some of the machine learning approaches we will be using require that the target values, which in this case is either KATEGORI_ID or UNDERKATEGORI_ID, are one a One-Hot Encoded format. If this is the case we One-Hot Encode the target values. This approach is also used to encode the industry codes we extract from the Brønnøysund registry.

Chapter 5

External Resources and Methods for Data Extraction

This section describes the external resources used in our research and the methods we implemented to extract data from them.

5.1 The Brønnøysund Entity Registry

The Brønnøysund Entity registry is a Norwegian governmental registry, accessible to the public, containing information about Norwegian companies. The registry includes information such as organization number, company address, business holder, and industry code. The industry code is a 2-part code represented as two numbers divided by a period. The first number represents the industry and the second part specifying the sub-category of said industry. This code is likely to be correlated with the categories mapped to the transaction descriptions. Therefore it is desirable to be able to extract the industry code for every transaction and use this to extend the feature set.

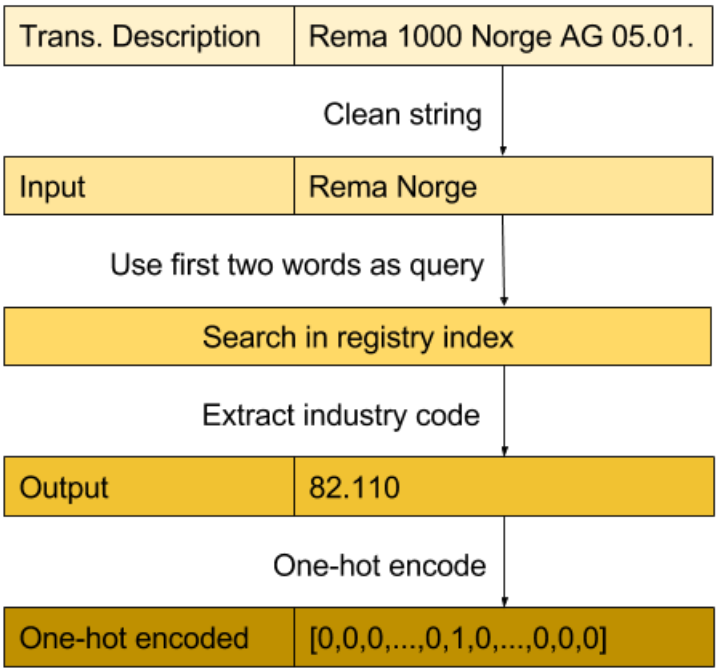

Figure 5.1: Industry Code Extraction Example

The *Brønnøysund Entity Registry* has an API through which its data is accessible. However, seeing as our system can only make around 2-10 requests per second against a REST API, it is beneficial to download the entire registry and index it manually. In our system, the registry is indexed using Whoosh, a fast, pure Python search engine library. In order to formulate search queries which will return relevant data, it is necessary to identify which part of the transaction description contains company information. It is only this part which should be used as search terms in the indexed entity registry. The transaction description is cleaned in the same way as described in Section 4.3 and the first two terms t_1 and t_2 in the resulting string are used to build the query Q

$$Q = t_1 \ ANDMAYBE \ t_2. \tag{5.1}$$

The ANDMAYBE operator means that we perform the query using t_1 and include t_2 if and only if a match is found while including it. Most of the time the first term describes the transaction well enough to make a successful lookup, but in some cases including the second term may be required. The system is now able to extract industry codes for transaction texts efficiently.

In order to put these codes in a format better suited for machine learning purposes, they are

one-hot encoded. They can then be appended to the bag-of-words feature set produced for the baseline. This is shown in Figure 5.2 b and c.

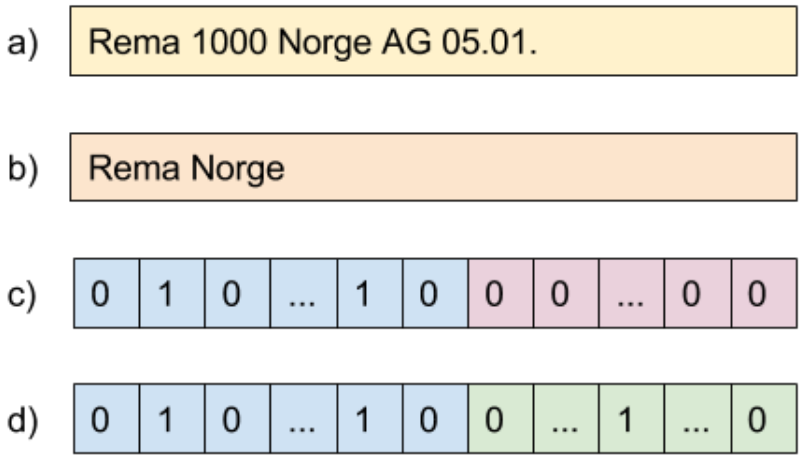

Figure 5.2: Transaction Representation Example
(a) Trans. text | (b) Trans.text Cleaned
(c) Bag-of-Words w/o Brreg Code
(d) Bag-of-Words with One-Hot Brreg Code

5.2 Google Places API

The *Google Places API Web Service*[1] is a service that returns information about places — defined within this API as establishments, geographic locations, or prominent points of interest — using HTTP requests. This Web Service allows for a particular type of query called Text Search Requests. This request service returns information about a set of places based on a string — for example, "pizza in New York" or "shoe stores near Ottawa" or "123 Main Street". The service responds with a list of places matching the text string, each of which contains a number of features. Among these features, there is a feature named 'types,' which is an array of feature types describing the given result.

[1] More information at https://developers.google.com/places/web-service/search

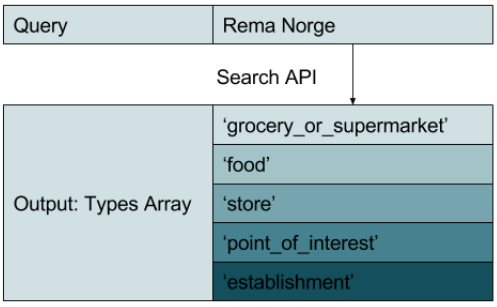

Figure 5.3: Google Places API Output Example

The types in this array are ordered according to specificity, meaning that the first entry is the most descriptive. An example of a *Google Places* types array is shown in Figure 5.3. These types are picked from a set of semantically defined types in the *Google Places* API. The first entry is extracted from this array and used as the type describing the transaction. There is likely to be some correlation between this type and the categories representing the transaction texts. It is therefore desirable to extract this data.

Seeing as this data is only accessible through the API and it costs a certain amount per request, it would not be financially or computationally sound to gather this information about every single transaction instance as done with the *Brønnøysund Entity Registry*. Therefore we have chosen a different approach where the subset of transactions which the classifier is not sufficiently confident about is identified and the *Google Places* data is collected for these transactions only.

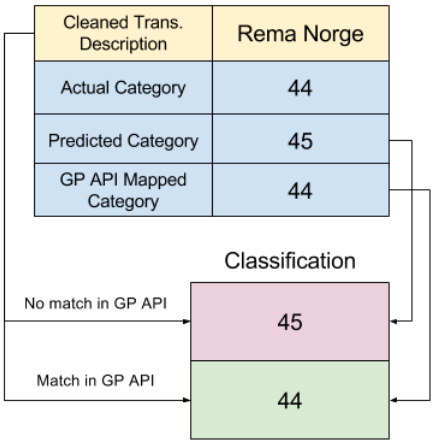

Figure 5.4: Google Places API Utilization Example

To identify this subset, the system evaluates the array of distances from the decision boundary for every class that the classifier produces for every transaction. If the distance measurement for a given class is positive, it means that the classifier predicts that the transaction belongs to this class. If it is negative, the classifier predicts it does not belong to the class. So, if there are multiple positive values in this array of distances, the classification model chooses the greatest one, but if there are none, the classification model is saying that the transaction doesn't belong to any of the classes. It is in this last case that we can conclude that the classifier is not sufficiently confident, and the *Google Places* approach used.

Of course, the classifier is not trained on the features gathered from the *Google Places* API so they cannot be added to the feature set to be used as input for the predictor. Therefore a direct mapping between *Google Places* type and transaction categories has been set up. Then, the system looks for a match for all of the non-confident classifications in the *Google Places* API. If there is a match, the mapping between *Google Places* Type and transaction category is used to decide the transaction's class. If there is no match, the system leaves the non-confident classification as it is.

This approach is exemplified in Figure 5.4 where a transaction with the description "Rema Norge" has been classified by the model to category 45. This classification is deemed non-confident, and a lookup is therefore made in the *Google Places* API. If this lookup results in a match, the classification will be changed to the category mapped to by the GP type extracted, which in this case is 44. If the lookup doesn't result in a match, the classification uses the original prediction of category 45. The *Google Places* approach does not handle classification to sub-categories. This is because the types employed in the *Google Places* API are not sufficiently descriptive to be mapped directly to sub-categories.

5.3 Wikidata and DBpedia

Wikidata and DBpedia are both Linked Open Data knowledge bases for extracting structured data from the web. Wikidata is a user curated source of structured information which is included in Wikipedia and DBpedia provides structured data from the Wikipedia and Wikimedia commons [3].

Both linked open data sources are structured in a hierarchy consisting of objects where their hierarchical relationships are described with RDF-triples. A RDF-triple contains three components; subject, predicate and object [11]. An example of a RDF-triple in Wikidata or DBpedia related to the project can be seen in Fig. 5.5.

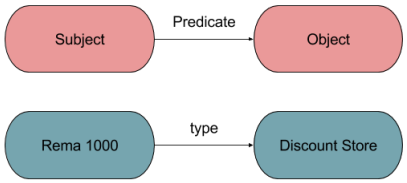

Figure 5.5: RDF-Triple Structure and Example

We want to acquire the meaning of the words in the transaction texts with the use of Wikidata and DBpedia. A visible trend in the original data is that a company name usually is present in the transaction texts so an approach would be to find the company name with the help of Wikidata and DBpedia, and find information of what industry the company operates in.

One API-call per transaction text would be very time consuming since our system can only do few API calls per second and the original data set is of size 220618. Since both Wikidata and DBpedia support queries through a SPARQL-endpoint that is capable of returning thousands of results and we are looking for something specific, a less time-consuming approach would be to find all companies and a description of what they do that Wikidata and DBpedia have structured data for and store it to a local file. We also do not need all the information that Wikidata and DBpedia can offer us about each company. An assumption of what would benefit training a prediction model the most would be a short description which specifically states something about what industry the company operates in. The closest predicate of which we could find that would fit our needs in Wikidata was *Description* and *Subjects* in DBpedia. The *Industry* predicate was considered and seemed more promising than *Subjects* in DBpedia but relatively few companies used this predicate unfortunately. The query results for Wikidata and DBpedia were stored in separate local files and the two were indexed to separate indexes with the company name as key and the description as value by using *Whoosh* [8] for quick and reliant look-up. The queries can be found in the Appendix 11.1.

From the companies in our index, we can find useful information about companies in the

transaction texts as seen in Fig. 5.6. First the transaction text is cleaned to remove all punctuation, numbers and words shorter than 3 letters. Then we search our index for the first and/or the second word in the transaction text which represent the company name and if a result is returned, we extend our transaction text. After this the process of Fig. 5.7 is applied and the new transaction text is converted to a Bag-of-Words representation. Depending on the information used to extend the original data is from the Wikidata index or the DBpedia index, the approaches are called the Wikidata approach or the DBpedia approach. If information is extended from both Wikidata and DBpedia the approach is called the Wikidata & DBpedia approach.

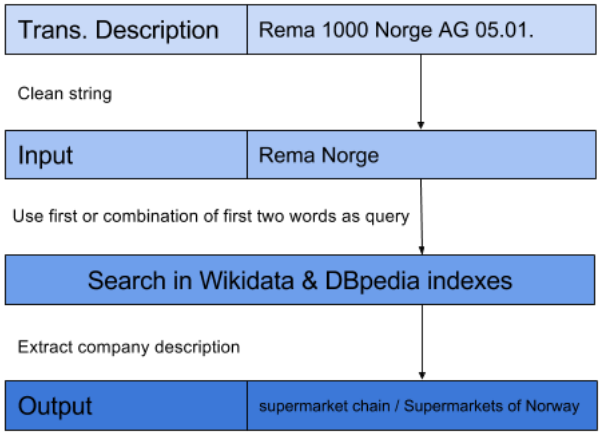

Figure 5.6: Wikidata and DBpedia Description Extraciton Example

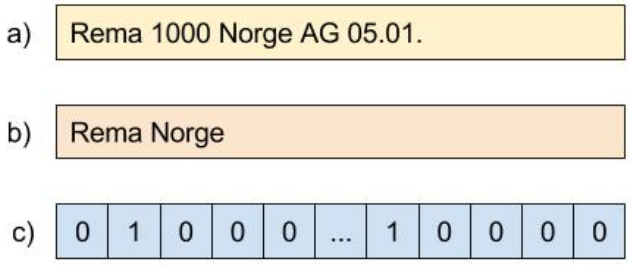

Figure 5.7: Transaction Representation Example
(a) Transaction Description
(b) Transaction Description Cleaned
(c) Bag-of-Words representation

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

5.4 WordNet

WordNet is a large lexical database of the English language and can be used for searching for definitions, synonyms and other information about English words [10]. It can also be used for simple translation from a supported language to English before doing a search. The information about the word will be returned in English.

Natural Language Toolkit [4] provides a module that can be downloaded so that WordNet is available locally. This means that no calls to an API are needed. This will make the process of searching for information about words much faster.

By using the WordNet module that *Natural Language Toolkit* provides, a word can be sent in and synonyms are returned if there are any (see Fig. 5.8).

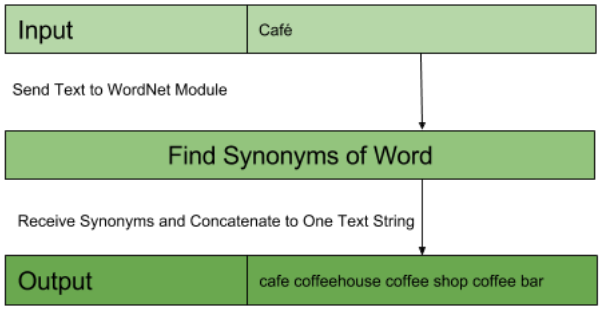

Figure 5.8: WordNet - Extraction of Synonyms for a Word

We are trying bring even more semantic meaning into the transaction texts by using synonyms. Often the same meaning is represented by using words that are synonyms. For instance, a café can be represented by the word coffee shop. By adding synonyms to a text we can create similarities between two texts that are initially viewed as dissimilarities since the words are written differently.

With the help of WordNet we intend to extend the transaction texts, and also the descriptions we receive from Wikidata and DBpedia, with synonyms so that similarities between two or more records that originally are not represented will be more transparent as they now share more words. As seen in (see Fig. 5.9) the data is first cleaned by removing punctuation, numbers, stopwords and words that are shorter than three and then split to get each word separate. Each word is sent to the WordNet module and the synonyms are returned. The words are then concatenated to one text string again. Data which i.e. contain the word 'bar' would now share

this word with a text which contains the word 'cafe'. The general similarities between two texts are now clearer after extending the data with synonyms. After extending the transaction text with synonyms the process in Fig. 5.7 is applied and the new transaction text is converted to a Bag-of-Words representation.

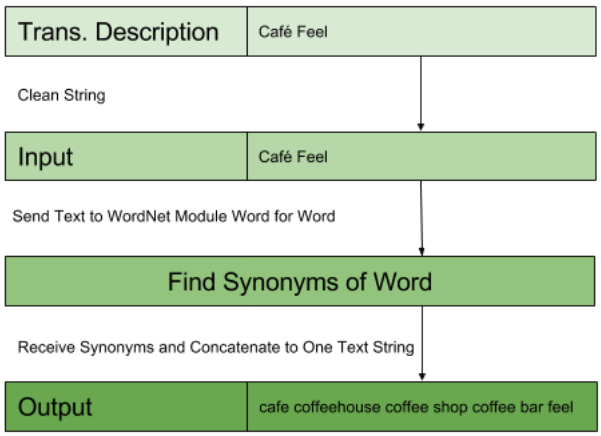

Figure 5.9: WordNet - Extraction of Synonyms for a Transaction

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

5.5 Yandex Translation

Yandex is a technology company that builds intelligent products and services powered by machine learning [1] and one of these services is a translation API that seem fit to translate the transaction texts in the original data set.

The transaction texts used in this project are in Norwegian and this could create problems when using the selected linked open data sources which returned descriptions are in English. By translating the original data to English we hope to compensate for the possible problems created by extending data with data on a different language. The translated transaction texts will hopefully share more words with the descriptions from the linked open data sources.

The Yandex Translate module can translate the original data word for word instead of translating the whole texts. This is done since it can be unfavourable to change the idiomatic meaning of the text and rather replace the individual words with their translations.

As shown in Fig. 5.10 the translation is extracted by first cleaning the text by removing punctuation, stopwords, numbers and and words shorter than 3 characters and then splitting the texts into separate words. We then translate each word respectively with the translation API. The returned translation of each word is then concatenated into one text string which constitute the new transaction text. The process of Fig. 5.7 is then applied and the translated transaction texts are then converted to a Bag-of-Words representation.

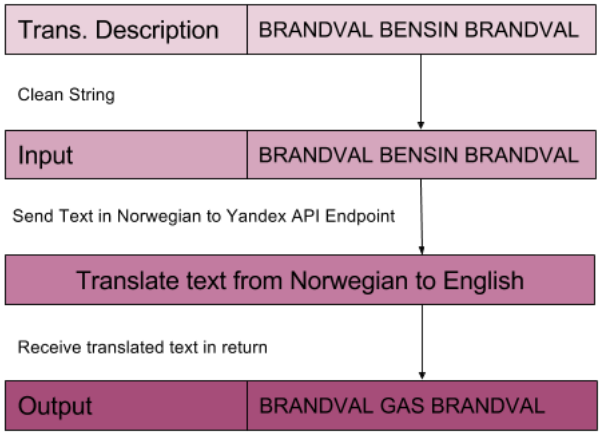

Figure 5.10: Yandex - Extraction of Translation for a Transaction

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

Chapter 6

Research Paper Outlines

This chapter gives brief overviews of the two papers written in conjunction with this thesis. The papers have been appended, in their entirety, to the end of this thesis.

6.1 Making Use of External Company Data to Improve the Classification of Bank Transactions

This project aims to explore to what extent external semantic resources on companies can be used to improve the accuracy of a real bank transaction classification system. The goal is to identify which implementations are best suited to exploit the additional company data retrieved from the *Brønnøysund Registry* and the *Google Places API*, and accurately measure the effects they have. The classification system builds on a Bag-of-Words representation and uses Logistic Regression as classification algorithm. This study suggests that enriching bank transactions with external company data substantially improves the accuracy of the classification system. If we compare the results obtained from our research to the baseline, which has an accuracy of 89.22%, the *Brønnøysund Registry* and *Google Places API* yield increases of 2.79pp and 2.01pp respectively. In combination, they generate an increase of 3.75pp.

6.2 Why Enriching Business Transactions with Linked Open Data May be Problematic in Classification Tasks

Linked Open Data has proven useful in disambiguation and query extension tasks, but their incomplete and inconsistent nature may make them less useful in analyzing brief low-level bank transactions. In this paper we investigate the effect of using Wikidata and DBpedia to aid in classification of real bank transactions. The experiments indicate that Linked Open Data may have the potential to effectively supplement transaction classification systems. However, given the nature of the transaction data used in this research and the current state of Linked Open Data sources, the extracted data has a negative impact the accuracy of the classification model when compared to the Baseline approach.

Chapter 7

Results

This section presents the experiments we have conducted and their results. We also give comparisons of the results of the different approaches.

7.1 Experiments

We have divided the experiments into four different sections:

- **Baseline Experiments**: Experiments conducted using the baseline approach
- **Research Paper Experiments Summaries**: A summary of the most important experiments conducted in the two research papers presented in Chapter 6.
- **Main Experiments**: Different experiments conducted in order to evaluate and improve our approaches.
- **Final Experiments**: Experiments conducted in order to evaluate our final approach. In this section, we use a combination of the approaches and parameter tuning which the experiments conducted in the previous sections have proven to be the most beneficial.

For every experiment the data set is divided into a training and test set, respectively 80% and 20% of the data set. All results are averaged over a given number of iterations, shuffling the training and test set each time. The number of iterations is specified in each experiment.

There is a total of 87199 distinct terms in the transaction texts. When we specify a Bag-of-Words size of X, this means using the X most frequently occurring terms.

We differentiate between *Main Categories* and *Sub Categories* as target values. This means that the target values used for training the model and performing the classifications are the main categories, of which there are 10, or the sub-categories, of which there are 63.

In our experiments we use a number of subsets of the **Original Data Set (ODS)**, sometimes referred to as the non-exclusive dataset, which is defined in Sec. 4. Since the sizes of these subsets are significantly smaller than ODS, we would also expect the results to be different. These subsets are defined as:

- **Wikidata Exclusive Subset (WES)** - This subset consists of only the bank transactions which yield a match in the Wikidata data source. The subset contains **113263** transactions and is used only in the Baseline, and Wikidata approaches.
- **DBpedia Exclusive Subset (DES)** - This subset consists of only the bank transactions which yield a match in the DBpedia data source. The subset contains **125765** transactions and is used only in the Baseline and DBpedia approaches.
- **Wikidata & DBpedia Exclusive Subset (WDES)** - This subset is the union of the Wikidata Exclusive Subset and the DBpedia Exclusive Data Set. The subset contains **136474** transactions and is used only in the Baseline, and Wikidata & DBpedia approaches.

In the *Brønnøysund Registry* approach we differentiate between "with" industry code and "exclusively" industry code. "With" means that all transactions are included, and the ones without a match in the registry are given a dummy value of 0 in place of the industry code. "Exclusively" means the system uses only the subset of transactions which have a match in the registry and therefore have a corresponding industry code. 192177 (87.13%) of the transactions in the dataset yield a match in the *Brønnøysund Entity Registry* thus constituting the "Exclusive" subset.

We also differentiate between two Multi-Class schemes for Logistic Regression; Using the 'liblinear' solver and using Softmax Regression. The 'liblinear' solver uses a coordinate descent algorithm and therefore does not learn a true multinomial model[7]. It uses a One-vs-Rest

scheme, meaning that a binary problem is fit for each label. Softmax Regression learns a multinomial model.

The evaluation metrics used are Accuracy (Micro-Averaged Recall), Macro-Averaged Recall, Macro-Averaged Precision and F-Score.

7.2 Results

7.2.1 Baseline Results

This experiment has been conducted with the following parameters:

- Bag-of-Words size of 4,000.
- Results averaged over 100 iterations.

Tables 7.1 shows the evaluation scores for the baseline approach using both multi-class schemes. Here we observe that the OvR scheme produces better Accuracy and Precision scores while the Softmax scheme yields higher Recall scores.

Multi-class Scheme	**Target Categories**	**Accuracy**	**Recall**	**Precision**	**F-Score**
One vs. Rest	Main Categories	0,8922	0,8668	0,9322	0,8951
One vs. Rest	Sub-categories	0,8632	0,7048	0,8934	0,7707
Softmax	Main Categories	0,8860	0,9280	0,8535	0,8853
Softmax	Sub-categories	0,8562	0,8749	0,6952	0,7574

Table 7.1: Baseline Results

Tables 7.2 and 7.3 show the per class evaluation scores for the OvR and Softmax multi-class schemes respectively.

Main Category	Precision	Recall	F-Score
42	0.96	0.88	0.92
43	0.94	0.87	0.90
44	0.98	0.92	0.95
45	0.76	0.96	0.85
47	0.88	0.81	0.85
48	0.93	0.74	0.83
49	0.93	0.83	0.88
103	0.96	0.81	0.88
104	0.99	0.88	0.93
181	0.99	0.98	0.98

Table 7.2: Baseline Per Class Results using One vs. Rest scheme.

Main Category	Precision	Recall	F1-Score
42	0.91	0.94	0.92
43	0.91	0.92	0.92
44	0.94	0.97	0.95
45	0.94	0.84	0.89
47	0.90	0.90	0.90
48	0.78	0.90	0.84
49	0.86	0.93	0.89
103	0.84	0.93	0.88
104	0.88	0.96	0.92
181	0.97	0.96	0.97

Table 7.3: Baseline Per Class Results using Softmax.

7.2.2 Research Paper Results

Research Paper 1: Making Use of External Semantic Resources

The experiments in this paper have been conducted with the following parameters:

- Bag-of-Words size of 4,000.
- Main Categories
- Logistic Regression with One vs. Rest Scheme.
- Results averaged over 100 iterations.

Table 7.4 show the evaluation results for the different approaches explored in the first paper.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	0,8922	0,8668	0,9322	0,8951
With Brreg Industry Codes	0,9201	0,8993	0,9395	0,9177
With Google Places Types	0,9123	0,8886	0,9369	0,9100
Combination	0,9297	0,9088	0,9426	0,9243

Table 7.4: Main results from the first research paper

Tables 7.5 and 7.6 show the per class evaluation scores for the *Brønnøysund Registry* and *Google Places* approaches respectively.

Main Category	Precision	Recall	F-Score
42	0.96	0.92	0.94
43	0.93	0.93	0.93
44	0.97	0.94	0.95
45	0.87	0.97	0.92
47	0.94	0.91	0.93
48	0.93	0.80	0.86
49	0.94	0.91	0.92
103	0.93	0.84	0.88
104	0.98	0.92	0.95
181	0.98	0.99	0.99

Table 7.5: Brønnøysund Registry Per Class Results.

Main Category	Precision	Recall	F-Score
42	0.97	0.90	0.93
43	0.94	0.90	0.92
44	0.97	0.93	0.95
45	0.81	0.97	0.89
47	0.92	0.88	0.90
48	0.93	0.74	0.83
49	0.94	0.87	0.90
103	0.94	0.83	0.88
104	0.98	0.91	0.94
181	0.99	0.98	0.98

Table 7.6: Google Places Per Class Results.

Table 7.7 shows a few key metrics pertaining to the *Google Places* approach. Count refers to the share of non-confident classifications. API Matches is the percentage of non-confident

classifications which yield a match in GP API. Positive is the share of API matches which map to correct class. The two last columns refer to the share of API matches which lead to positive and negative alterations of the classification.

Count	API Matches	Positive	False -> Positive	Positive -> False
13.94%	65.60%	43.99%	23.68%	1.98%

Table 7.7: Google Places Key Metrics.

Every class is affected by positive and negative classification changes. If we normalize the percentages of negative and positive classification changes for each class by taking those values and multiplying them by their corresponding weights in Table 7.7, respectively 0.2368 and 0.0198 for positive and negative classification changes, we get a measure of how much each class is affected by the classification changes. Calculating the difference between these yields a value which indicates whether or not the approach contributes positively (> 0) or negatively (< 0) towards the accuracy of the class. This is shown in Table 7.8.

Norm. Positive Class Change	Norm. Negative Class Change	Class contribution (Diff.)
3,50	0,24	3,26
4,63	0,15	4,48
0,35	0,24	0,11
3,16	0,67	2,50
6,22	0,25	6,00
0,95	0,14	0,81
4,13	0,26	3,87
0,15	0,02	0,13
0,31	0	0,31
0,27	0,03	0,24

Table 7.8: Per Class Classification Change Contribution

Research Paper 2: Making Use of Linked Open Data

The experiments in this paper have been conducted with the following parameters:

- Bag-of-Words size of 4,000
- Main Categories

- Softmax Regression
- Results averaged over 10 iterations.

The approaches in this section are extended and/or enhanced variants of the baseline (see Sec. 7.2.1) which means that the original data have been altered or appended to.

Wikidata

Table 7.9 shows the model's evaluation scores after the descriptions from Wikidata have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	88.60%	92.80%	85.35%	88.53%
Wikidata	84.65%	89.85%	81.56%	84.92%
Wikidata + Translation	84.99%	88.97%	82.01%	84.90%
Wikidata + Translation + Synonyms	79.87%	85.27%	76.23%	79.79%

Table 7.9: Wikidata Approach Results on the Original Data Set

Table 7.10 shows the model's evaluation scores after the descriptions from Wikidata have been used to extend the data in the Wikidata Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	94.39%	94.13%	90.57%	92.17%
Wikidata	92.69%	92.19%	87.89%	89.77%
Wikidata + Translation	93.15%	92.11%	88.33%	89.98%
Wikidata + Translation + Synonyms	91.83%	90.96%	87.04%	88.72%

Table 7.10: Wikidata Approach Results on the Wikidata Exclusive Subset

DBpedia

Table 7.11 shows the model's evaluation scores after the descriptions from DBpedia have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	88.60%	92.80%	85.35%	88.53%
DBpedia	86.48%	90.65%	83.74%	86.58%
DBpedia + Translation	86.65%	89.85%	84.20%	86.59%
DBpedia + Translation + Synonyms	81.74%	86.19%	78.33%	81.46%

Table 7.11: DBpedia Approach Results on the Original Data Set

Table 7.12 shows the model's evaluation scores after the descriptions from DBpedia have been used to extend the data in the DBpedia Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	94.26%	94.65%	90.67%	92.46%
DBpedia	93.48%	93.21%	89.23%	90.99%
DBpedia + Translation	93.53%	92.87%	89.24%	90.84%
DBpedia + Translation + Synonyms	92.41%	91.54%	87.46%	89.19%

Table 7.12: DBpedia Approach Results on the DBpedia Exclusive Subset

Combination of Wikidata & DBpedia

Table 7.13 shows the model's evaluation scores after the descriptions from Wikidata and DBpedia have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	88.60%	92.80%	85.35%	88.53%
Wikidata & DBpedia	82.97%	88.55%	79.48%	83.02%
Wikidata & DBpedia + Translation	83.48%	87.86%	79.73%	82.95%
Wikidata & DBpedia + Translation + Synonyms	79.71%	84.63%	75.27%	78.70%

Table 7.13: Wikidata & DBpedia Approach Results on the Original Data Set

Table 7.14 shows the model's evaluation scores after the descriptions from Wikidata and DBpedia have been used to extend the data in the Wikidata & DBpedia Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Baseline	94.08%	94.40%	90.70%	92.39%
Wikidata & DBpedia	92.13%	92.01%	87.98%	89.75%
Wikidata & DBpedia + Translation	92.42%	91.53%	87.64%	89.33%
Wikidata & DBpedia + Translation + Synonyms	91.13%	90.29%	86.24%	87.94%

Table 7.14: Wikidata & DBpedia Approach Results on the Wikidata & DBpedia Exclusive Subset

7.2.3 Main Experiments Results

Bag-of-Words Size Experiment

This experiment has been conducted with the following parameters:

- Baseline approach only - No external data sources.
- Main Categories.
- Logistic Regression with One vs. Rest Scheme.
- Results averaged over 100 iterations.

Table 7.15 shows the evaluation scores for the baseline for different Bag-of-Words sizes.

Bag-of-Words Size	Accuracy	Recall	Precision	F-Score
1000	0,7887	0,7453	0,8733	0,7919
2000	0,8543	0,8237	0,9110	0,8595
3000	0,8789	0,8515	0,9256	0,8829
4000	0,8926	0,8673	0,9328	0,8956
5000	0,9005	0,8759	0,9375	0,9028
6000	0,9056	0,8824	0,9408	0,9081
7000	0,9100	0,8881	0,9433	0,9125
8000	0,9132	0,8919	0,9452	0,9156
9000	0,9162	0,8958	0,9471	0,9187
10000	0,9188	0,8995	0,9485	0,9214
11000	0,9208	0,9019	0,9497	0,9233
12000	0,9225	0,9041	0,9507	0,9250
13000	0,9245	0,9063	0,9519	0,9269
14000	0,9257	0,9077	0,9526	0,9280
15000	0,9268	0,9089	0,9530	0,9289
16000	0,9277	0,9101	0,9532	0,9296
17000	0,9284	0,9108	0,9535	0,9302
18000	0,9293	0,9114	0,9535	0,9306
19000	0,9300	0,9120	0,9538	0,9310
20000	0,9305	0,9126	0,9538	0,9312

Table 7.15: Bag-of-Words Size Experiment Results

Table 7.16 shows the increase in accuracy between every 1000 increment in Bag-of-Words size. Here we observe that the accuracy delta drops below 0.1% after Bag-of-Words size of 15,000.

Bag-of-Words Size	**Δ Accuracy**
2000	0,0656
3000	0,0245
4000	0,0137
5000	0,0079
6000	0,0051
7000	0,0044
8000	0,0032
9000	0,0030
10000	0,0025
11000	0,0020
12000	0,0018
13000	0,0020
14000	0,0012
15000	**0,0010**
16000	0,0009
17000	0,0007
18000	0,0010
19000	0,0006
20000	0,0006

Table 7.16: Bag-of-Words Δ Accuracy

Incremental Training Set Size Experiment

This experiment has been conducted with the following parameters:

- Baseline approach only - No external data sources.
- Main Categories.
- Logistic Regression with One vs. Rest Scheme.
- Results averaged over 100 iterations.

Figure 7.1 shows the evaluation scores for different subset sizes of the original training set.

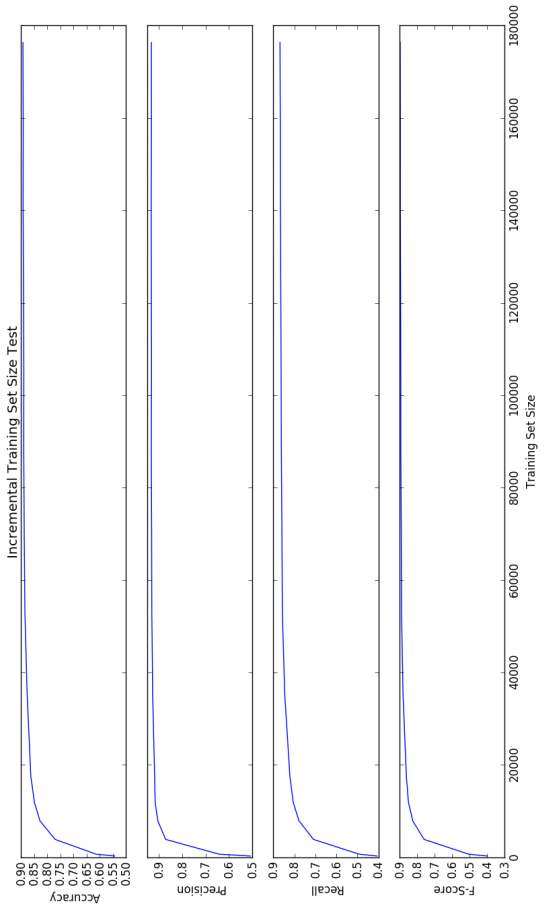

Figure 7.1: Incremental Training Set Size Experiment Results

Brreg Code Imputation Results

This experiment has been conducted with the following parameters:

- Bag-of-Words size of 4,000.
- Main Categories
- Softmax Regression
- Results averaged over 10 iterations.

Table 7.17 shows the results of the *Brønnøysund Registry* approach with and without imputation of industry codes. In order to impute the missing industry codes, we have used a logistic regression classifier trained on the transactions for which we can extract industry codes. Without imputation means we have used the baseline approach and a "dummy"-value of zero has been used for missing industry codes. Here we observe a decline in the evaluation scores after implementing the imputation.

Approach	**Accuracy**	**Recall**	**Precision**	**F-Score**
Without Imputation	0.9163	0.9337	0.8951	0.9128
With Imputation	0.9091	0.9217	0.8899	0.9046

Table 7.17: Results with and without *Brønnøysund Registry* Industry Code Imputation

Feed-Forward Neural Network Evaluation Results

This experiment has been conducted with the following parameters:

- Bag-of-Words size of 4,000.
- Main Categories
- FFNN with Softmax output layer
- Results averaged over 10 iterations.

Table 7.18 shows the performance scores of our implementation of a FFNN classifier. The number of-of hidden layers L is defined as

$$L = A \tag{7.1}$$

where $\{A \in \mathbb{Z} | A > 0\}$, and the number of hidden neurons N is defined as

$$N = \lfloor (B * I) + O \rfloor \tag{7.2}$$

where $\{B \in \mathbb{R} | 0 <= B <= 1\}$, I is the size of the input layer, and O is the size of the output layer.

The size of the input and output layers are constant; I = 4000 and O = 10.We examine four different configurations based on the rules of thumb introduced in section 2.9.

A	**B**	**L**	**N**	**Accuracy**	**Recall**	**Precision**	**F-Score**
1	1/3	1	1343	0.8892	0.9192	0.8631	0.8876
1	2/3	1	2676	0.8905	0.9185	0.8655	0.8888
2	1/3	2	1343	0.8923	0.9189	0.8666	0.8896
2	2/3	2	2676	0.8917	0.9187	0.8659	0.8891

Table 7.18: Results from using FFNN as Classifier

Ensemble Approach Results

This experiment has been conducted with the following parameters:

- Bag-of-Words size of 4,000.
- Main Categories
- Logistic Regression with One vs. Rest Scheme.
- Results averaged over 100 iterations.

An ensemble approach means using a combination of models in order to increase performance. In this case this means using a Brreg model trained exclusively on transactions for which we have an industry code to classify transactions with an industry code, and using a Baseline model trained on all transactions to classify transactions without an industry code.

Table 7.19 shows accuracy scores for test subsets with and without industry codes. They show the accuracy scores for the baseline approach when trained on the entire data set and the accuracy scores for the *Brønnøysund Registry* approach when trained on the industry code subset. The percentages next to the subset labels indicate the size of the subset.

Subset	**Baseline**	**Brreg Approach**
Exclusively with industry code (87.1%)	0.8918	0.9380
Exclusively without industry code (12.9%)	0.7988	Not applicable

Table 7.19: Subset Accuracies for Ensemble approach

Table 7.20 shows the accuracy scores for the ensemble approach. We observe that this approach yields the same accuracy score as when using only the *Brønnøysund Registry* approach.

Subset	**Ensemble Accuracy**
Entire Dataset (100%)	**0.9201**
Exclusively with industry code (87.1%)	0.9380
Exclusively without industry code (12.9%)	0.7988

Table 7.20: Ensemble Approach Accuracy Scores

Human Classifier Experiment

We have performed an experiment where we had two people manually classify random samples of 200 transactions. They achieved an average accuracy of 93%, which indicates that the transaction descriptions are not always sufficiently descriptive.

7.2.4 Final Experiment Results

The experiments in this section have been conducted with the following parameters:

- Brønnøysund Registry and Google Places approaches applied for main category results.
- Brønnøysund Registry approach applied for sub-category results.
- Bag-of-Words size of 15,000.
- Logistic Regression with One vs. Rest Scheme.
- Results averaged over 100 iterations.

Target Categories	**Accuracy**	**Recall**	**Precision**	**F-Score**
Main Categories	0,9464	0,9320	0,9554	0,9430
Sub-Categories	0,9174	0,8079	0,9130	0,8478

Table 7.21: Results for final approach

Chapter 8

Discussion

This chapter is aimed at discussing our final results. To open this section, we will briefly introduce what we identify as the most important topics of discussion. We will begin by discussing the different classification algorithms we have applied, and interpret and compare the results they yielded. We will then proceed to discuss the results produced with regards to the *Brønnøysund Registry* and *Google Places* approaches both individually and combined. We will explain what their significance is and discuss what implications they may have. By doing this, we hope to shed light on what we believe to be the reasons behind why we got the results that we did. Furthermore, we will discuss the Linked Open Data source approaches and analyze why these led to a decline in performance measures. To conclude this section, we will discuss what decisions we made to build our final classification model and evaluate the results this model produced.

8.1 Choice of Classification Algorithm

We chose to explore the use of Logistic Regression as classification algorithm because we believe our data to be linearly separable. Linear models are robust and tend to need much less hand holding than more sophisticated approaches [17]. Seeing as we are dealing with a multiclass classification problem, we needed to explore different ways of handling this. We ended up pursuing a One vs. Rest scheme and the multinomial approach using Softmax Regression. The One vs. Rest scheme fits a binary classifier to each class and chooses the classifier which maximizes the likelihood of a transaction belonging to a given class. Conversely, the multinomial

classifier learns all the classes directly. In this way, the parameters of each class are estimated interdependently and the model built may be more robust against outliers.

Figure 8.1: Comparison of classification Algorithm Results

We also decided to try a Feed-Forward Neural Network by adding a number of hidden layers to the softmax regression model. We did this to see whether or not we could better separate our data using a more complex model. We tried some different configurations as shown in Table 7.18 and observed only non-significant differences in results. A comparison of the results obtained using the baseline approach in combination with our three classifiers is shown in Fig 8.1. For the FFNN we used the best configuration of hidden layers and neurons. The first thing we see is that adding hidden layers to the Softmax Regression classifier leads to a minimal change in the evaluation scores. We can, therefore, conclude that our data does not need a more complex model than a linear one and using the FFNN is superfluous and increases the likelihood of overfitting.

We also observe that the OvR method scores the highest in accuracy, meaning that it correctly classifies the most transactions. The macro-averaged recall scores are however significantly better for the Softmax method, while the macro-averaged precision scores are significantly better for the OvR method. If we look at tables 7.2 and 7.3, which show the per class

results of the OvR and Softmax classification methods, we observe that the recall score for the OvR method is very high (96%) for class 45, which is the class with the greatest support. The precision is also quite low for this class (76%). This indicates that employing the OvR scheme results in the classifier having a slight bias towards class 45. For the Softmax method, this bias is eliminated, and the recall score for class 45 drops to 84% and increases for all other classes, thus increasing the macro-averaged recall but decreasing the accuracy and macro-averaged precision. One could argue that the softmax approach is the better approach because it yields better recall across the line for the different classes, but ultimately we want a classification model which correctly classifies the maximum amount of transactions. We, therefore, accept the slight reduction in recall for the other classes and conclude that the Logistic Regression with a One vs. Rest scheme is the best approach.

8.2 External Semantic Resources

Table 8.1 shows the improvement in evaluation scores each of the approaches in this section made in relation to the Baseline.

Approach	**Accuracy**	**Recall**	**Precision**	**F-Score**
Brønnøysund Registry	**2,79 %**	**3,25 %**	0,73 %	2,26 %
Google Places	**2,01 %**	**2,18 %**	0,47 %	1,49 %
Combination	**3,75 %**	**4,20 %**	1,04 %	2,92 %

Table 8.1: Percentage point improvements

8.2.1 *The Brønnøysund Registry*

The intuition behind utilizing the industry codes extracted from the *Brønnøysund Entity Registry* was that there would be some correlation between these and the target values for our transactions. This led to the hypothesis that using industry codes to extend our feature set would give rise to an increase in the accuracy of our classification model. Our results show an increase in accuracy of 4.58 and 2.79 percentage points respectively for the exclusive and non-exclusive methods of evaluating the approach. Exclusive here referring to using only the subset of our data for which we were able to extract industry codes.

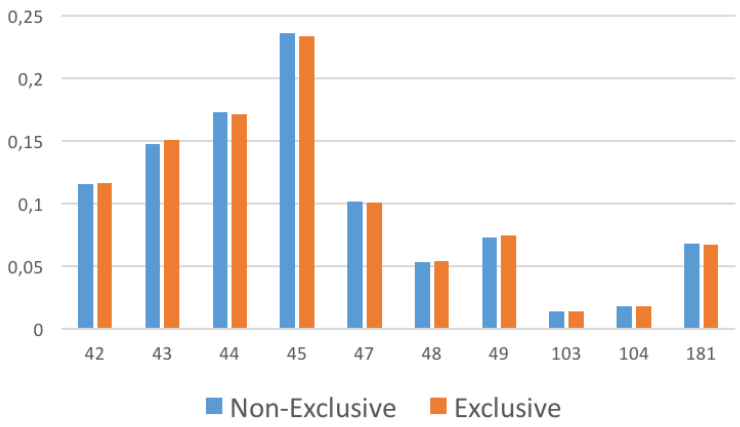

Figure 8.2: Comparison of label distributions for data set and subset with industry codes

The gap in accuracy between the exclusive and non-exclusive evaluation approaches may have occurred for two possible reasons. The first is that the exclusive subset has a distribution of transactions which are more easily classified. The second reason could be that when using the exclusive subset, the classifier is not affected by the 'dummy' value which is assigned to all transactions without a corresponding industry code. When we assign this 'dummy' value, we are telling the classifier that all the transactions with this value have something in common, when in reality they may have nothing in common.

To identify which of these two reasons contribute to the gap in accuracy, we look at the label distributions for the exclusive and non-exclusive transaction sets shown in Figure 8.2. These are approximately the same, indicating that the baseline results should be roughly the same in both cases. However, if we compare the per class results for the *Brønnøysund Registry* approach in Table 7.5 and the Baseline in Table 7.2, we see that the former performs better for the larger classes (43, 44, and 45). This could explain the gap in accuracy since the transactions without industry codes are not diminishing the effects of the *Brønnøysund Registry* approach in the exclusive subset. In other words, this indicates that replacing missing industry codes with a 'dummy'-value is the factor which causes this accuracy gap between the exclusive and non-exclusive transaction sets.

The ideal situation would be to have industry codes for all transactions, but we are only able to retrieve industry codes for approximately 87% of all transactions. We, therefore, decide to use the 'dummy'-values and accept the loss in contributed accuracy from the *Brønnøysund Registry*

approach.

Another way to mitigate the loss in contributed accuracy from the *Brønnøysund Registry* caused by using the 'dummy'-values would be to use some imputation technique for the missing industry codes. We did this using a logistic regression classifier trained on the transactions with industry codes. In essence, we made a classifier for industry codes. As we can see in Table 7.17 this approach leads to a decline in accuracy. There are 620 different industry codes in our dataset. We believe this makes the data too difficult to separate and the imputed Brreg codes are a source of error, and therefore lead to a decline in accuracy.

The *Brønnøysund Registry* approach adds minimal overhead to the running time of the system. This is because it has been downloaded and indexed, and therefore can be queried locally. The downside to this approach is that the index is not kept up to date automatically. As we can see in both the Baseline and *Brønnøysund Registry* results, the evaluation scores suffer a significant decline when classifying to the sub-categories. This is because the complexity of separating the data increases with the number classes.

8.2.2 *Google Places* API

This approach is a post-processing technique which aims to identify classifications which are believed to be incorrect and try to reclassify them in order to increase the accuracy of the system. The approach identifies 13.94% of the classifications as non-confident. These are the classifications which the system will try to reclassify by searching for a match in the *Google Places* API. Of these classification instances, we can find a match in the GP API for 65.6% of them, and 43.99% of these result in a correct classification. This means that as a stand-alone classifier it would achieve an accuracy score of approximately 28% (product of the number of matches and number of correct classifications), which is very poor.

If there is a match for a given transaction in the *Google Places* API, this approach can have four outcomes:

- False -> Positive: GP mapping changes incorrect prediction to correct.
- False -> False: GP mapping changes incorrect prediction to same or other incorrect prediction.

- Positive -> Positive: GP mapping leaves prediction unchanged.
- Positive -> False: GP mapping changes correct prediction to incorrect.

We refer to these outcomes as classification changes. The Positive-to-Positive and False-to-False classification changes are not interesting as they will have no effect on the accuracy of the system. It is desirable to maximize the False-to-Positive classification changes as these will increase accuracy, and minimize Positive-to False-classification changes as these will decrease accuracy. As we can see in Table 7.8 the class contributions are positive for all classes meaning that positive classification changes outnumber the negative classification changes in all classes. If this were not the case, we could omit certain classes from the *Google Places* approach to increase its efficiency.

Ultimately, the *Google Places* approach leads to a 2.01 percentage point increase in accuracy compared to the baseline. It is, however, a time-consuming procedure as we are required to make requests to a REST API for all non-confident classifications.

8.2.3 Combining the *Brønnøysund Registry* and *Google Places* Approaches

When we combine the two approaches discussed in this paper, we would expect to reap the benefits of both approaches. This is almost the case, but there is a slight overlap between the two approaches when it comes to which transactions they improve the accuracy for. The classes where there is no overlap the contribution in accuracy from the two approaches separately should equal the contribution of the approaches in combination. If their combined contribution is smaller than the sum of individual contributions, then there is an overlap in the transactions they correctly classify.

If we look at table 8.2 we can see the difference between combined contribution and sum of individual contributions defined as the overlap measure. If the overlap measure is 0, there is no overlap, if it is negative its magnitude determines the amount of overlap in the class. We observe that six of the ten of the classes are affected by this overlap.

Main Category	Sum indiv. approach	Combined approach	Overlap Measure
42	0,04	0,05	-0,01
43	0,06	0,08	-0,02
44	0,02	0,02	0
45	0,01	0,01	0
47	0,1	0,14	-0,04
48	0,06	0,06	0
49	0,08	0,09	-0,01
103	0,03	0,05	-0,02
104	0,04	0,06	-0,02
181	0,01	0,01	0

Table 8.2: Per class overlap measure between approaches. The second column shows the sum of the improvements contributed by the two approaches individually. The third column shows the improvement contributed by the approaches in combination. The final column shows the overlap measure.

8.3 Linked Open Data

8.3.1 Linked Open Data as a Resource

As we see in Table 8.3 the results produced using the Wikidata and DBpedia approaches show a performance decline compared to the Baseline approach. The observed results indicate that the Baseline approach itself was better suited for training a classification model than the proposed approaches experimented with was. The accuracy of the Baseline approach was **88,60%** and by using the Wikidata, DBpedia, and Wikidata & DBpedia approaches we can observe from Table 8.4 a decline in accuracy of **3,95%**, **2,12%** and **5,63%**. We also notice a corresponding drop in the other performance measures. As we have shown with the approaches in the previous section and the research presented in 3, it is indeed possible to improving accuracy using feature enrichment techniques. Expanding the feature set allows the classifier to find more distinct patterns on which to make decisions. Unfortunately, this was not the case with the data we collected from Wikidata and DBpedia. By further analysis of each linked open data source, we discuss possible justifications for our results.

Baseline	Wikidata	DBpedia	Wikidata & DBpedia
88.60%	84.65%	86.48%	82.97%

Table 8.3: Accuracy for The Baseline, Wikidata, DBpedia and Wikidata & DBpedia Approaches with the Original Data Set

Wikidata	DBpedia	Wikidata & DBpedia
-3.95%	-2,12%	-5,63%

Table 8.4: Accuracy change of the Linked Open Data Approaches from the Baseline on the Original Data Set

First and foremost, the hit-ratio, denoting how many transactions yielded a match in the linked sources, was relatively small. By counting the number of transactions that produced a result in each linked open data source we noticed that only a little over half of the original data yielded a hit in Wikidata and DBpedia:

- Wikidata Hit-Rate = $\frac{113263}{220618} = 51,34\%$
- DBpedia Hit-Rate = $\frac{125765}{220618} = 57,01\%$
- Wikidata & DBpedia Hit-Rate = $\frac{136474}{220618} = 61,86\%$

Combining the Wikidata and DBpedia approaches was an attempt to increase this hit-rate, but still yielded a relatively low number. The reason for this was the great amount of overlap in which transactions yielded a match in the data sources. There were as many as **102554** transactions in the original data which yielded a result in both Wikidata and DBpedia. Only **10709** of the transactions were found exclusively in Wikidata and **23211** transactions were found exclusively in DBpedia.

The low hit-rates of all three approaches indicate that the linked open data sources are not extensive enough, separate or combined, for our use and are not likely to contribute positively when training our classification model.

Having observed these low hit-rates, we conducted experiments where we used the subsets of the original dataset which contained only transactions which yielded a match in the linked open data sources. We did this to gain insight into how the linked open data approaches could

potentially perform compared to the Baseline approach given that all of the original data yielded a match in the linked open data sources.

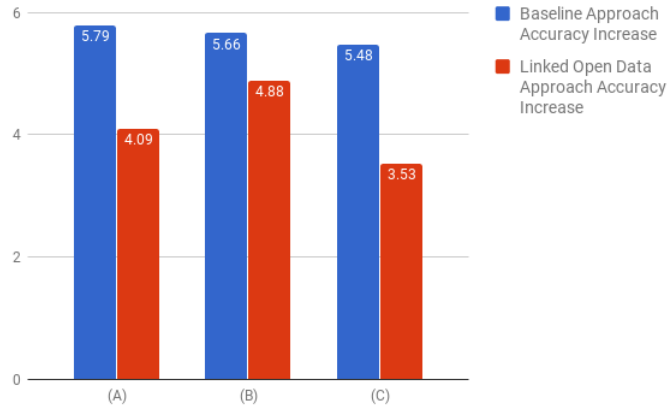

Figure 8.3: Accuracy Comparison from the Original Data Set to a Reduced Original Data Set
(A) Comparison of accuracy percentage change of Baseline approach and the Wikidata approach from using the Baseline approach Original Data Set to the Wikidata Exclusive Subset.
(B) Comparison of accuracy percentage change of Baseline approach and the DBpedia approach from using the Original Data Set to the DBpedia Exclusive Subset.
(C) Comparison of accuracy percentage change of Baseline approach and the Wikidata & DBpedia approach from using the Original Data Set to the Wikidata & DBpedia Exclusive Subset.

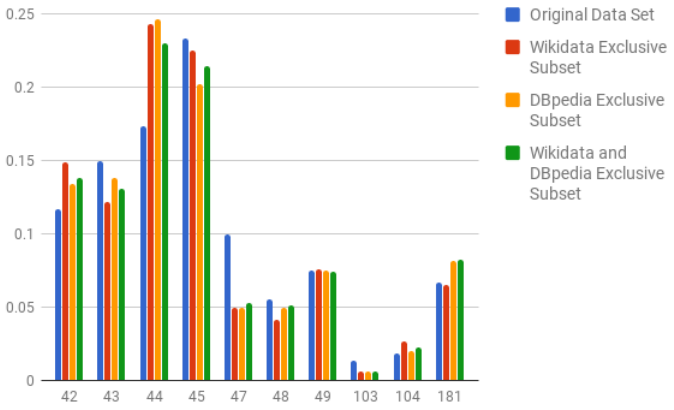

Figure 8.4: Normalized Label Distribution of Different Data Sets

As we can see in Figure 8.3, all linked open data approaches increased substantially to the better. However, the performance was still worse in all of the linked open data approaches than in the Baseline approach. This shows that much of the error from using the linked open data sources is introduced by the fact that a lot of the original data does not yield a result in the

linked open data sources. We had a theory that the increase in performance could be explained by the subsets having a label distribution which favored classes with a higher recall score. We can, however, see from the label distribution in Fig. 8.4 that the label distribution, with the exception of the classes **44** and **47**, is approximately the same for the original data set and its subsets. We could, therefore, conclude that the performance increase rather indicates that the removed transactions within each class were harder to classify.

The low hit-rate could be explained by the nature of the bank transaction texts. As stated in 4, our dataset consists of Norwegian transaction texts where many contain Norwegian company names. This makes it more difficult for us to get results from Wikidata and DBpedia since they contain relatively few Norwegian companies. Both Wikidata and DBpedia are focused on a more general level which makes deeper knowledge on a specific topic hard to obtain from them e.g. companies on a country basis. Smaller companies that operate in only one country are, understandably, not a priority when covering information on a global scale. On the other hand, larger companies and companies that are internationally known tend to give results even though they may be based in only one country. The information that can be extracted from Wikidata and DBpedia seems to be too general for the purpose of this project and does not give information to the extent that we require.

A side-effect of Wikidata and DBpedia covering information on a more general global basis is that the returned information might not represent the correct information. By this we mean that many results are *False Positives* which would make the information we extend the original data with incorrect and misleading. By conducting a Simple Random Sample test for each linked open data source we could see an indication of this. Each Simple Random Sample test consisted of 100 transactions which yielded a result in each of the linked open data sources. The evaluation was done as a subjective analysis since there is no actual correct answer to this test and therefore results may or may not represent the true results. The sample data of the tests revealed this for each linked open data source:

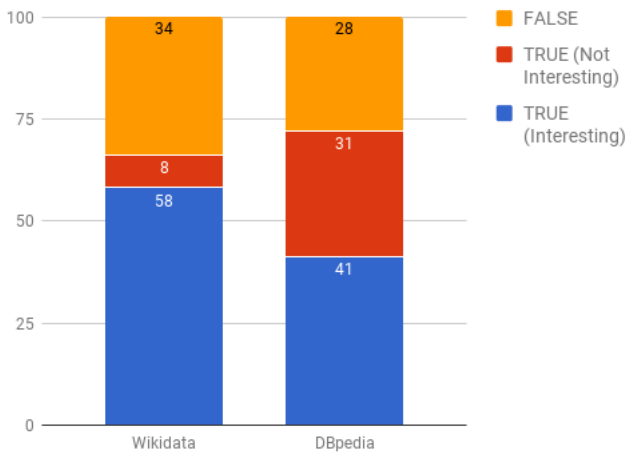

Figure 8.5: Comparison of Hit-Value of Wikidata and DBpedia

If we assume that this Simple Random Sample test is representative of the rest of the data that comes from Wikidata and DBpedia, then this is a clear indication that we are introducing many words which do not describe the transactions they are assigned to. We should then expect to observe a performance decline in both the results for Wikidata (see Table 7.9) and DBpedia (see Table 7.11) approaches compared to the results for the Baseline approach. This is also shown in the performance decline in the experiments performed on the Wikidata (see Table 7.10) and DBpedia (see Table 7.12) subsets when compared to the Baseline on each respective data set.

From the Simple Random Sample, we see that Wikidata yields a higher percentage of correct and meaningful results than DBpedia since a lot of the results from DBpedia give little meaning. From this perspective, we would believe that Wikidata approach would perform better than the DBpedia approach. However, as we observe in the results, this theory does not hold up, and DBpedia performs a little better. This indicates that extending the transaction descriptions with data that does not contribute to distinguishing between classes, produces better performance results. This suggests that the data from the linked open data sources do not help in this classification problem. These results could also explain why the combined approach performed worse than both the Wikidata and DBpedia approaches because even if one of the linked open data sources return a correct result, the other one may return an incorrect result.

The data returned from both Wikidata and DBpedia was of variable length and content. Two companies that operate in the same industry would often have a description that was written

differently, and no standard format was used. This could be another possible source of error. Conformity could have been an advantage for classification since a decision boundary would be more pronounced in the data. The description from Wikidata and DBpedia mainly consists of free-text which makes the description of many companies that operate in the same industry highly variable. This error could also be thought to make the performance of the combination of the two approaches to decline even further, which we believe is another reason for the poor result.

8.3.2 Correction of the Proposed Approaches

In order to remedy the shortcomings of the linked open data sources, we have used two methods in an attempt to correct this. First, we translate the original data to English and then extend both the transaction descriptions and the data from Wikidata and DBpedia with synonyms.

The translation of the original data showed improvement as seen from the result of all approaches (see Tables 7.9, 7.11 and 7.13). We believe that this increase in performance come from the reason that translated original data share more words with the extracted linked open data than with the original data itself. We can observe this effect from the increase in performance when the experiment is conducted on both the original data set and the reduced original data set. The performance increase is true for all proposed approaches. However, even though the there was an increase, the difference was not significant. As seen in the change from Table 8.4 to Table 8.5, the observed improvement obtained by using the translated original data instead of just the original data for the proposed approaches was **0,34%**, **0,17%** and **0,51%** for Wikidata, DBpedia and the combined approaches respectively.

Wikidata	**DBpedia**	**Wikidata & DBpedia**
-3.61%	-1,95%	-5,12%

Table 8.5: Accuracy change of the Linked Open Data Approaches with Translated Original Data from the Baseline on the Original Data Set

Extending the translated original data with synonyms in addition to Wikidata and DBpedia did however not result in a performance increase. As seen in Table 8.6 the performance in the experiments is clearly reduced. We believe that the way synonyms were used to extend the dif-

ferent approaches further contributes to making the problems observed even more significant by looking at the data returned by Wikidata and DBpedia. When we extend with synonyms to create similarities we also, as a side-effect, further reduce the conformity of the transaction texts. This is an effect created by adding many new words to the transaction texts. We also believe that since the data returned might be incorrect, the synonyms only enhance the observed error and therefore also create errors of greater significance. This side-effect was not taken into account when selecting approaches. For these reasons we can see that extending the translated original data and linked open data with synonyms only contribute to a less clear decision boundary to perform classifications on.

Wikidata	**DBpedia**	**Wikidata & DBpedia**
-8,73%	-6,86%	-8,89%

Table 8.6: Accuracy change of the Linked Open Data Approaches with Translated Original Data and Synonyms from the Baseline on the Original Data Set

A proposition for a better correction approach would be to replace words rather than just adding them to the text. If word ***A*** and word ***B*** are synonyms, replace them with a word ***C***. See Fig. 8.6 for an example.

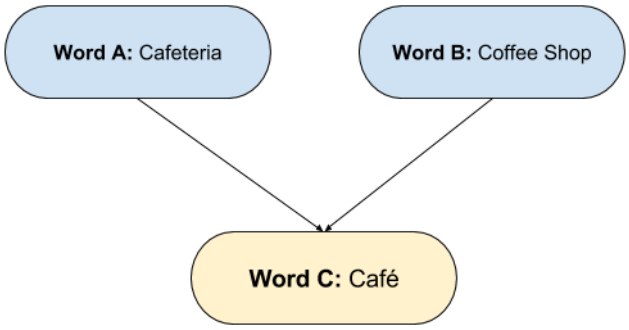

Figure 8.6: An Example of Replacing Synonyms with the Same Word

Using the translation approach may also have this effect since it is a possibility that synonyms could be translated to the same word. A more strict filtering method for choosing which words to find synonyms for could also be beneficial since many of the synonyms we extended contributed to confusion when finding a pattern of which we make classifications based on.

However, the correction would most likely only result in a small increase in performance since the data of which we extract from Wikidata and DBpedia still is insufficient for use in this project.

8.4 Approach Comparison

We wish to briefly discuss why the External Semantic Resource approaches using the *Brønnøysund Registry* and *Googles Places* API improve the accuracy of the system where Linked Open Data fails to do so. Firstly, the *Brønnøysund Registry* is well suited for our data because it is an exclusively Norwegian registry. It is also all-encompassing meaning that all Norwegian companies which fulfill certain criteria are required to be registered. All the data in this registry is semantically defined, and it returns matches for 87.1% of our transaction descriptions

In the Linked Open Data (LOD) sources we have used, there are no requirements or incentives to register, and few companies consider it a priority. This makes the LOD sources inadequate when it comes to which companies it contains information about. The Linked Open Data sources only yield matches for around 50% of our transactions, and as shown in Figure 8.5 a large portion of these are erroneous matches or matches which do not contain any valuable information.

Also, the data which is recorded about every company in the *Brønnøysund Registry* is highly structured. Every company contains a pre-defined set of structured features, and for our purpose, this includes the industry code which is highly correlated with the targets classes of our system. For the Linked Open Data sources, this is not the case. There is a low degree of conformity between the company entries, and the features are often unstructured. This means that even if the LOD sources were to yield more matches, there is no guarantee that they will all contain a feature which describes them in a comparable way. For the LOD approaches, this results in an inadequate basis for feature generation for our model, which in turn means that they are not able to supplement the data with any meaningful patterns.

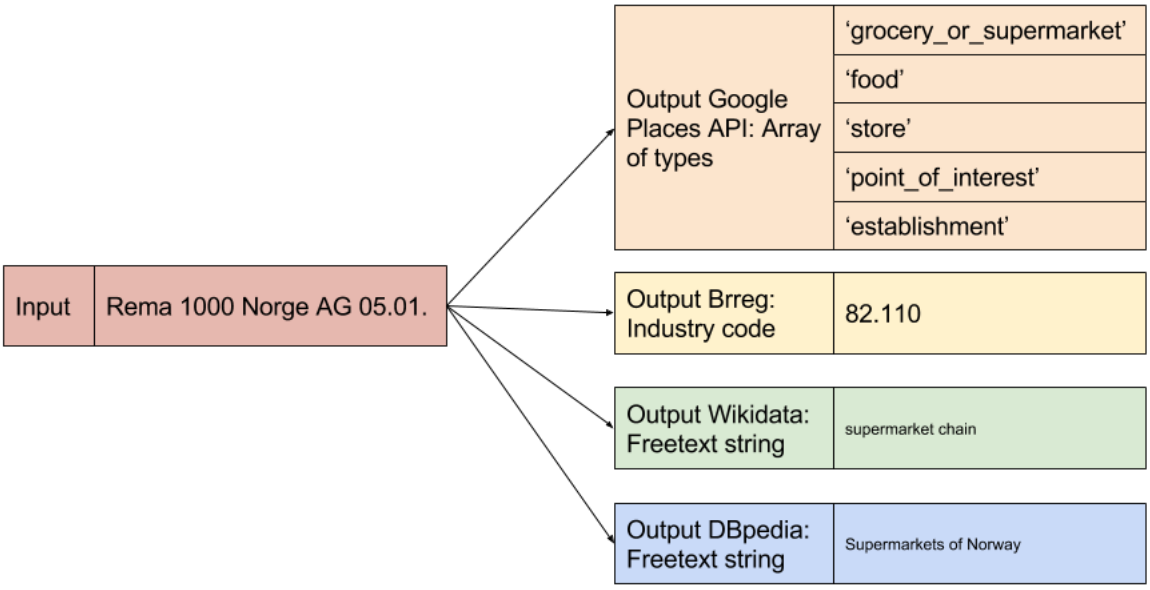

Figure 8.7: Output examples

An example of actual outputs from a string which matches in all four external resources is shown in Figure 8.7. The outputs from the *Brønnøysund Registry* and the *Google Places API* are taken from a set of semantically predefined values, while the output from the LOD resources are free text. This means that a lookup in the Linked Open Data sources for another Norwegian supermarket may return an entirely different text and there will be no information which links the two together. Norwegian supermarkets should will however always return the same industry code and types from the *Brønnøysund Registry* and the *Google Places API*, meaning that they make a much better contribution to separating original data.

8.5 Final Approach

Our final classification model uses Logistic Regression with a One vs. Rest multi-class scheme. We based this decision on the evaluation scores produced by the different classification algorithms. However, in order to build our final model we had to make a few key decisions. When it came to deciding which external semantic resources we would use, the choice was easy - we would use only a combination of those approaches which resulted in an increase in accuracy. That is, the *Brønnøysund Registry* approach and the *Google Places approach*.

Initially, we worked with a Bag-of-Words size of 4,000. If we look at Figure 8.8, which plots the accuracy scores of the classifier against different Bag-of-Words sizes, this begins to converge at around 4,000.

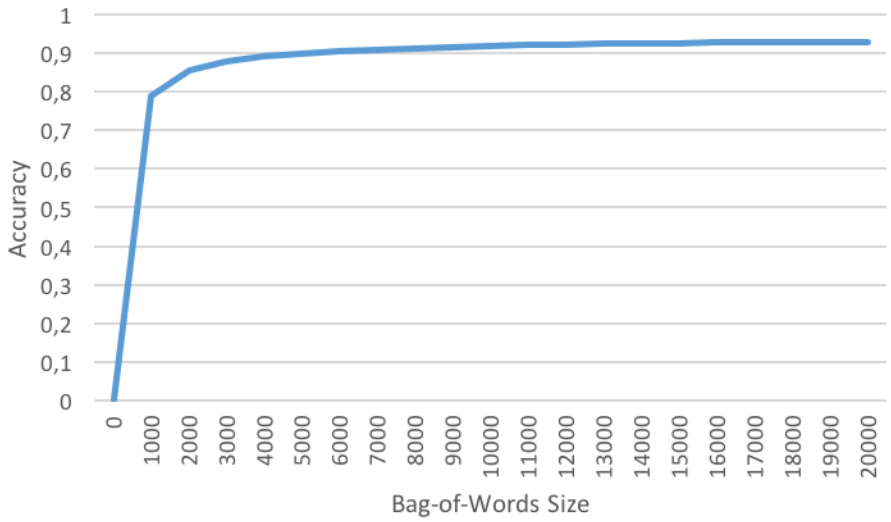

Figure 8.8: Accuracy per 1000 increment in Bag-of-Words size

However, there are still significant increases in accuracy over the ten next 1,000 increments in Bag-of-Words size. We, therefore, decided to set a threshold of 0.1% for this accuracy delta; when the accuracy delta dropped below this threshold, we set the Bag-of-Words size. If we look at Table 7.16 this accuracy delta drops below 0.1% after the Bag-of-Words size exceeds 15,000. Therefore we use a Bag-of-Words size of 15,000 in our final model.

As discussed in the section covering the *Brønnøysund Registry* approach, there is a small but significant increase in accuracy when exclusively classifying transactions with *Brønnøysund Registry* industry codes. In order to exploit this, we proposed using an ensemble of models where each model is trained to handle exclusively on transactions for which we found a match a corresponding external semantic resource. When implementing this we only used the baseline approach and the *Brønnøysund Registry* approach seeing as the models using *DBpedia* and *Wikidata* performed poorer than the baseline. The subset of transactions which yield a match in the *Brønnøysund Registry* constitutes 87.1% of the total transactions. Our hypothesis was that all transactions with *Brønnøysund Registry* codes would be classified using the exclusive Brreg model with an accuracy of 93.8% and the ones without would be classified us-

ing the Baseline approach with an accuracy of 89.22%, thus resulting in a total accuracy of $93.8 * 0.871 + 89.22 * 0.129 = 93.21\%$. However, if we look at Table 7.20 we see that it results in a total accuracy of 92.01% which is the same as when using the standalone *Brønnøysund Registry* approach. Table 7.19 indicates that this is because the baseline approach performs proportionally worse on the subset for which we do not find matching *Brønnøysund Registry* matches. In practice, there is, therefore, no advantage to using this approach.

We also considered the fact that our classifier could be subject to overfitting. This could happen if our training set did not contain sufficient training examples for all the classes. As we can see in Figure 7.1 the evaluation metrics converge at around 50,000 training examples. Our training set consists of 180,000 transactions, thus making this a robust foundation for training our model.

Our final approach yielded an accuracy of 94.46%, and seeing as our human classifier experiment resulted in an average accuracy of 93% we can argue that our data does not provide enough information for classification methods to achieve evaluation scores that are much higher than this.

Chapter 9

Conclusions

In this chapter we present a summary of the most important findings in the project. We also give our recommendations for future work, and present changes and extensions which we believe have the potential to improve the classification system. All conclusions will be conveyed in relation to the research questions we introduced in section 1.3.

Research Question 1: *How do Logistic Regression and a Feed-Forward Neural Network compare in the classification of bank transactions?*

Firstly, Feed Forward Neural Networks are computationally expensive. Back-propagating through a number of hidden layers with large amounts of neurons takes time. This classification model is intended to be used in a real-time application and will have requirements to meet when it comes to running time and computational efficiency. In addition our results show that the FFNN provides no significant improvements when compared with the far simpler approach Logistic Regression. This shows that our data is linearly separable and a Logistic Regression classifier is well suited to cover our needs. Also, the FFNN classifier introduces a far larger risk for overfitting.

We have investigated two multi-class schemes for Logistic Regression, both of which prove to score well on all performance measures. The One-vs-Rest scheme we investigated scores higher on macro-averaged precision and overall accuracy, while the multinomial scheme scores higher

for macro-averaged recall. Seeing as we wish to maximize the number of transactions correctly classified, we decided that the One-vs-Rest scheme was the best approach to pursue.

Research Question 2: *How can external semantic resources like Brønnøysundregisteret and Google Places be used to improve the accuracy of the classification system?*

The research we have conducted on these two external semantic resources shows that they can be used both as data sources for feature enrichment techniques as well as for post-processing techniques. The *Brønnøysund Registry* was used for the former and the *Google Places* API for the latter, both yielding significant improvements in performance scores. The *Brønnøysund Registry* approach was, however, the best contributor, both regarding an increase in accuracy and running time as it requires very little overhead compared to the *Google Places* approach. The two approaches work well in combination yielding a total increase in accuracy of 3.75%. These techniques can be transferred and applied to text classification problems in other domains. The challenge lies in determining which external resources should be used. A multi-label solution to this approach, and data, would also be a potentially useful area to study.

Research Question 3: *Can linked open data sources like DBPedia and WikiData be used to improve the accuracy of the classification system?*

Firstly, we can conclude that Wikidata and DBpedia are not fit to be use as data sources in the classification of bank transactions. For a domain like this, consistency and conformity are critical. Due to the nature of linked open data, the granularity of the searches in the linked open data sources is important to get useful information of which we can use to extend the bank transactions with. We found that by using Wikidata and DBpedia, we get very few results on specific domains like businesses, primarily Norwegian, and too much information which either was too

lacking or too descriptive to make improvements in our classification problem. A finding in this research, however, is that the data which is possible to extract from Wikidata and DBpedia is better suited for internationally known companies. This means that the results potentially could have been better if the experiments were conducted on a different data set where the companies mainly were internationally known companies and not country specific companies.

Despite our two attempts to correct the shortcomings of the data extracted from both linked open data sources, the yielded results from the attempted approaches were still inferior to the Baseline approach. We believe that with a better approach of how to make use of synonyms we could have produced better results, although, still limited by the quality of the linked open data.

The concept of a structured web is interesting, and using all of this available information shows potential. If the linked open data sources continue to grow and conformity is introduced to the structured data then linked open data may prove useful in projects like this in the future.

Chapter 10

Acronyms

BRREG Brønnøysund Registry

GP Google Places

LOD Linked Open Data

API Application Programming Interface

LR Logistic Regression

OvR One-vs-Rest

FFNN Feed-Forward Neural Network

Chapter 11

Appendix

11.1 SPARQL queries

These are the SPARQL queries used to extract data from the linked open data sources Wikidata and DBpedia. *LIMIT N* denotes how many results is returned from a query and *OFFSET M* denotes from where in the results the query should start retrieval. Used in succession with different *N* and *M*, they can retrieve all results of the query.

11.1.1 Wikidata

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?name ?itemDescription
WHERE {
    ?item wdt:P31/wdt:P279* wd:Q4830453 .
    ?item rdfs:label ?name .
    SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en" .
    }
}
```

```
LIMIT N
OFFSET M
```

11.1.2 DBpedia

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?company ?subjects {
    {
        SELECT ?company ?subjects {
            ?company a/rdfs:subClassOf* dbo:Company .
            OPTIONAL {?company dct:subject ?subjects .}
        }
        ORDER BY ?company
    }
}
LIMIT N
OFFSET M
```

11.2 Categories & Subcategories

KATEGORI_ID	KATEGORI_NAVN	UNDERKATEGORI_ID	UNDERKATEGORI_NAVN
42	Bil og transport	80	Bompenger og parkering
42	Bil og transport	82	Frakt og varetransport
42	Bil og transport	78	Taxi
42	Bil og transport	76	Offentlig transport
43	Bolig og eiendom	67	Elektrisk utstyr og hvitevarer
43	Bolig og eiendom	69	Alarm og sikkerhet
43	Bolig og eiendom	71	Maskin og industri
43	Bolig og eiendom	73	Strøm og oppvarming
43	Bolig og eiendom	65	Interiør, møbler og belysning
43	Bolig og eiendom	66	Jernvare og byggevare
44	Dagligvarer	63	Diverse dagligvarer
45	Opplevelse og fritid	103	Reiseselskap
45	Opplevelse og fritid	105	Diverse opplevelse og fritid
45	Opplevelse og fritid	102	Kino, kultur og arrangementer
47	Helse og velvære	106	Optikk
47	Helse og velvære	108	Apotek og helsekost
47	Helse og velvære	111	Diverse helse og velvære
48	Hobby og kunnskap	93	Aviser og magasiner
48	Hobby og kunnskap	95	Utdanning og opplæring
48	Hobby og kunnskap	90	Bøker, musikk og film
48	Hobby og kunnskap	99	Diverse hobby og kunnskap
48	Hobby og kunnskap	91	Leker, spill og hobby
49	Klær og utstyr	88	Barn
49	Klær og utstyr	86	Tur og sport
49	Klær og utstyr	84	Klær, sko og tilbehør
103	Annet	116	Avgifter
103	Annet	117	Taxfree
103	Annet	114	Juridiske tjeneste
104	Kontanter og kredittkort	141	Kontantuttak
181	Finansielle tjenester	123	Gebyr
181	Finansielle tjenester	120	Sparing
42	Bil og transport	83	Diverse bil og transport
42	Bil og transport	81	Toll og veiavgift
42	Bil og transport	79	Verksted, service og utstyr
42	Bil og transport	77	Bensinstasjon
42	Bil og transport	75	Bil, båt og motor
43	Bolig og eiendom	68	Telefon, TV og internett
43	Bolig og eiendom	70	Håndverker
43	Bolig og eiendom	72	Borettslag og sameie
43	Bolig og eiendom	64	Blomster og planter
43	Bolig og eiendom	74	Diverse bolig og eiendom
44	Dagligvarer	62	Kioskvarer
44	Dagligvarer	61	Mat og husholdning
45	Opplevelse og fritid	104	Trening og fritidsaktiviteter
45	Opplevelse og fritid	101	Hotell og overnatting
45	Opplevelse og fritid	100	Restaurant, kafé og bar
47	Helse og velvære	110	Personlig pleie
47	Helse og velvære	107	Helsetjeneste
47	Helse og velvære	109	Frisør
48	Hobby og kunnskap	94	Veldedighet
48	Hobby og kunnskap	96	Kontorrekvisita
48	Hobby og kunnskap	98	Dyrehold
48	Hobby og kunnskap	92	Kunst og foto
48	Hobby og kunnskap	97	Tipping og pengespill
49	Klær og utstyr	87	Renseri og reparasjon
49	Klær og utstyr	85	Smykker og klokker
49	Klær og utstyr	89	Diverse klær og utstyr
103	Annet	118	Barnehage
103	Annet	115	Skatt
104	Kontanter og kredittkort	113	Kredittkort
104	Kontanter og kredittkort	112	Minibank
181	Finansielle tjenester	121	Lån
181	Finansielle tjenester	122	Forsikring

Table 11.1: Categories, Subcategories and their IDs

Bibliography

[1] About yandex. Available at https://yandex.com/company/.

[2] Ontologies. Available at https://www.w3.org/standards/semanticweb/ontology.

[3] Wikidata and dbpedia. Available at http://wikidata.dbpedia.org/. Last accessed 10/06/2017.

[4] (2017). Natural language toolkit. Available at http://www.nltk.org. Last accessed 13/06/2017.

[5] Albitar, S., Espinasse, B., and Fournier, S. (2014). Semantic enrichments in text supervised classification: application to medical domain. *Florida Artificial Intelligence Research Society Conference*.

[6] Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, Cambridge, Massachusetts, London, England, 2nd edition.

[7] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ.

[8] Chaput, M. (2012). About whoosh. Available at http://whoosh.readthedocs.io/en/latest/intro.html#about-whoosh. Last Accessed 09/06/2017.

[9] Espinosa, R., Garriga, L., Zubcoff, J. J., and Mazón, J.-N. (2014). Linked open data mining for democratization of big data. *IEEE International Conference on Big Data*.

[10] Fellbaum, C. (2005). Wordnet and wordnets. *Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier*, pages 665–670.

[11] Group, R. W. (2014). Resource description framework (rdf). Available at `https://www.w3.org/RDF/`. Last accessed 29/05/2017.

[12] Gutiérrez, Y., Vázquez, S., and Montoyo, A. (2011). Sentiment classification using semantic features extracted from wordnet-based resources. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 139–145.

[13] Heaton, J. (2008). *Introduction to Neural Networks for Java*. Heaton Research, Inc, second edition.

[14] Iftene, A. and Baboi, A. (2014). Using semantic resources in image retrieval. *20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, 96:436–445.

[15] Lausch, A., Schmidt, A., and Tischendorf, L. (2015). Data mining and linked open data – new perspectives for data analysis in environmental research. *Ecological Modelling 295. Elsevier*, pages 5–17.

[16] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. The MIT Press, Cambridge, Massachusetts, London, England.

[17] Perlich, C. (2016). Which is your favourite machine learning algorithm? Available at `http://www.kdnuggets.com/2016/09/perlich-favorite-machine-learning-algorithm.html`. Last accessed 14/06/2017.

[18] Poyraz, M., Ganiz, M. C., Akyokus, S., Gorener, B., and Kilimci, Z. H. (2012). Exploiting turkish wikipedia as a semantic resource for text classification. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–5.

[19] Prud'hommeaux, E. and Seaborne, A. (2008). Sparql query language for rdf. Available at `https://www.w3.org/TR/rdf-sparql-query/`. Last accessed 07/06/2017.

[20] Raman Arora, Amitabh Basu, P. M. and Mukherjee, A. (2016). Understanding deep neural networks with rectified linear units. Available at `https://arxiv.org/pdf/1611.01491.pdf`. Last accessed 15/05/2017.

[21] Skeppe, L. B. (2014). Classify swedish bank transactions with early and late fusion techniques. Master Thesis. KTH Royal Institute of Technology, Stockholm.

[22] Van Asch, V. (2013). Macro- and micro-averaged evaluation measures. Available at `http://www.clips.uantwerpen.be/~vincent/pdf/microaverage.pdf`. Last accessed 16/05/2017.

[23] W3C. What is linked data. Available at `https://www.w3.org/standards/semanticweb/data`. Last accessed 01/06/2017.

[24] Xiong, C. and Callan, J. (2015). Query expansion with freebase. *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, September 27-30, Northampton, Massachusetts, USA*.

[25] Y., Y., F., M., and Rong H., H. J. (2004). Improved email classification through enriched feature space. *Advances in Web-Age Information Management*.

Making Use of External Company Data to Improve the Classification of Bank Transactions

Erlend Vollset[1], Eirik Folkestad[1], Marius Rise Gallala[2], and Jon Atle Gulla[1]

[1] Department of Computer Science,
Norwegian University of Science and Technology.
[2] Sparebank1 SMN

Abstract. This project aims to explore to what extent external semantic resources on companies can be used to improve the accuracy of a real bank transaction classification system. The goal is to identify which implementations are best suited to exploit the additional company data retrieved from the *Brønnøysund Registry* and the *Google Places API*, and accurately measure the effects they have. The classification system builds on a Bag-of-Words representation and uses Logistic Regression as classification algorithm. This study suggests that enriching bank transactions with external company data substantially improves the accuracy of the classification system. If we compare the results obtained from our research to the baseline, which has an accuracy of 89.22%, the *Brønnøysund Registry* and *Google Places API* yield increases of 2.79pp and 2.01pp respectively. In combination, they generate an increase of 3.75pp.

Keywords: Classification, Bank Transactions, Logistic Regression, Semantic Resources

1 Introduction

This project has been carried out in collaboration with Sparebank1 in order to gain insight into the classification of bank transactions. Progress in the domain at the intersection of finance and machine learning is important as it has plenty of potential applications; accurate consumption statistics, financial trend predictions, and fraud detection to name a few. In the research that we have previously conducted on automatic classification of bank transactions [7], we have developed a system which utilizes transaction description texts to classify transactions. This approach has proven to be somewhat effective with a classification accuracy of 89%, but we wish to develop techniques to improve this approach further by enriching our feature set using external semantic resources.

We examine two external semantic resources; the *Brønnøysund Entity Registry*, containing information about Norwegian companies, and the *Google Places API*, containing information about businesses, companies, and establishments worldwide. Two main approaches to the problem are covered:

- Using extracted external data to extend the baseline feature set
- Using extracted external data to aid in the classification of transactions where the classifier is not sufficiently confident.

This paper gives a detailed description of the implementation of these two approaches. It also provides a thorough analysis of the results obtained from testing the system. We compare the results to a baseline in order to draw meaningful conclusions about the impact of the approaches studied. Due to the general nature of the techniques in this project, they can easily be transferred to other applications within text classification. Seeing as they have shown to improve the accuracy of the system, they introduce a new dimension to problem-solving in the classification domain.

The remainder of this paper is structured as follows. Section 2 describes the theoretical foundation upon which we have built our project. It explains in detail the techniques we have implemented, as well as giving a detailed description of the data we have used and how it is represented. Section 3 follows with providing a presentation of the experiments we have conducted, as well as the results we obtained from them. In Section 5 we present a few studies which are closely related to the work we are conducting in this project. The paper is summarized in sections 4 and 6 by discussing our findings, providing recommendations for further work, and drawing our final conclusions.

2 Data and Methods

2.1 Data set

The bank transaction data set consists of 220619 unstructured Norwegian transaction descriptions. These are actual bank transactions from a given time interval provided to us by Sparebank1 SMN, the central Norway branch of Sparebank1. SpareBank1 is a Norwegian alliance and brand name for a group of savings banks. The alliance is organized through the holding company SpareBank1 Gruppen AS that is owned by the participating banks. In total the alliance is Norway's second largest bank and the central Norway branch is the largest bank in its region.

Table 1: Transaction entry example

Description	Sub-category	Main Category
Rema 1000 Norge AG 05.01.	61	44

Table 2: Main Categories and their IDs

ID	Main Category Name	Category Name English
42	Bil og transport	Automobile and Transport
43	Bolig og eiendom	Housing and Real-Estate
44	Dagligvarer	Groceries
45	Opplevelse og fritid	Recreation and Leisure
47	Helse og velvære	Health and Well Being
48	Hobby og kunnskap	Hobby and Knowledge
49	Klær og utstyr	Clothes and Equipment
103	Annet	Other
104	Kontanter og kredittkort	Cash and Credit
181	Finansielle tjenester	Financial Services

Each transaction description in the data set is labeled with a corresponding category and sub-category. There is a total of 10 main categories and 63 sub-categories. The main categories are shown in table 2. An example of an entry in the dataset is shown in Table 1.

We have also performed a human classifier experiment where we had two people manually classify random samples of 200 transactions. They achieved an average accuracy of **93%**, which indicates that the transaction descriptions are not always sufficiently descriptive. This limits the evaluation scores we should expect the system to yield.

2.2 Bag-of-Words Model

We continue this section by introducing a few concepts essential to understanding the approaches we have implemented. The Bag-of-Words Model is used to convert the transaction descriptions to a representation better suited for machine learning. This particular technique is commonly used in natural language processing and information retrieval. In our application of the model, it is used as a tool for feature generation. When generating features for a corpus of texts, each text is represented as a multiset (bag) of the terms contained in the text. Given a corpus of texts $X = x_1, x_2$, where

x_1 = Alan has a chair
x_2 = A chair is a chair

the bag-of-words representation produced is shown in Figure 2.a. The resulting matrix has a column for each term in the corpus and a row for each text. The value is the term frequency, i.e., the number of occurrences of the term in a given text. These features may then be used as input to a predictive model such as the one in this project.

X	Alan	has	a	chair	is
x1	1	1	1	1	0
x2	0	0	2	2	1

(a) Bag-of-Words

C1	1	0	0
C2	0	1	0
C3	0	0	1

(b) One-Hot Encoding

Fig. 2: Representation Examples

2.3 One-Hot Encoding

One-Hot is a sequence of bits where a single bit is 1, and the rest are 0. One-Hot Encoding is a method for representing a set of features using One-Hot bit sequences. The length of the sequence of bits is equal to the size of the set of features. The bit which represents the given feature is 1 and all others 0. Assume three categories denoted as C_1, C_2, and C_3, their One-Hot encoded representation is shown in Figure 2.b.

The feature being represented is projected onto a plane, and all the produced planes are in equal distance of each other. This categorical representation ensures

that there is no ordinal relationship between the features. This makes it ideal for representing non-numerical features. We have used this technique to represent certain external data elements.

2.4 Logistic Regression

In this project, we have used the Logistic Regression algorithm implemented in the Scikit-Learn machine learning library for Python. This is a linear algorithm and estimates the probability of a class A given a feature-vector B. It does this by applying a logistic function to find the relationship between the class and the feature vector. It assumes that the distribution P(A|B), where A is the class and B is the feature-vector, is on a parametric form and then estimates it using the training data. The probability P(A|B) of B belonging to class A is given by the sigmoid function (see Eq. 1 and Eq. 2).

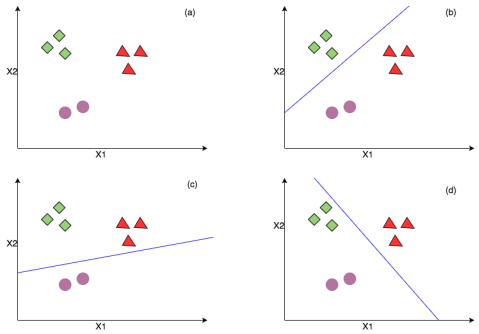

Fig. 3: Logistic Regression OvR example (a) feature-vectors | (b) classifier for diamonds | (c) classifier for circles | (d) classifier for triangles

P(A|B) is estimated by creating linear combinations of the features of X and multiplying them by some weight w_i and applying a function $f_i(A|B)$ on the combinations. f_i returns a value denoting the relationship between a feature of a class and a feature in a feature-vector based on the probability exceeding a certain threshold. This value is either true or false. The weight w_i denotes the importance of the feature.

$$z(A,B) = \sum_{i=1}^{N} w_i f_i(A,B) \tag{1}$$

$$P(A|B) = \frac{1}{1 + exp(-z(A,B))} \tag{2}$$

This classifier uses a discriminative algorithm which means that it can compute $P(X|Y)$ directly, without having to compute the likelihood of $P(Y|X)$ first. From Logistic Regression's discriminative properties it can be assumed that it has a small asymptotic error compared to the generative approaches. However, it requires a larger set of training data to achieve such results.

In our implementation, we use the 'liblinear' solver provided by scikit-learn. This solver uses a coordinate descent algorithm and therefore does not learn a true multinomial model[1]. Instead, it uses a One-vs-Rest scheme, meaning that a binary classifier is trained for each class. These classifiers predict whether or not

an observation belongs to the class. Then, to classify new observations, you pick the class whose classifier maximizes the probability of the observation belonging to it. In subfigures (a), (b), and (c) in Figure 3, data from each individual class has been fit to their respective classifiers.

2.5 Baseline

A baseline refers to a set of techniques and configurations applied to our system intended to serve as a basis for defining change and measuring improvement. In our system, the baseline approach is a standard machine learning approach to text classification which involves using a Bag-of-Words representation and Logistic Regression. We have chosen to use this model because we believe our data to be linearly separable. Also, linear models are robust and tend to need much less hand holding than more sophisticated approaches [4].

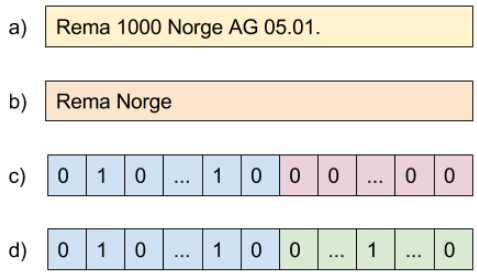

Fig. 4: Transaction Representation Example
(a) Trans. text | (b) Trans.text Cleaned
(c) Bag-of-Words w/o Brreg Code
(d) Bag-of-Words with One-Hot Brreg Code

A number of preprocessing steps are applied to the data in order to prepare it for the classification algorithm. First, the description string is cleaned to remove all punctuation, numbers, and words shorter than three letters (see Fig. 4b). The text is then converted to a vector representation using the Bag-of-Words Model (see Fig. 4c).

2.6 Brønnøysund Entity Registry

The *Brønnøysund Entity Registry* is a Norwegian governmental registry, accessible to the public, containing information about Norwegian companies. The registry includes information such as organization number, company address, business holder, and industry code. This industry code is likely to be correlated with the categories representing the transaction descriptions. Therefore it is desirable to be able to extract this industry code for every transaction and use this to extend the feature set used as input to the classification model. Seeing as the data is semantically defined, we can automate this lookup.

The *Brønnøysund Entity Registry* has an API through which its data is accessible. However, seeing as our system can only make around 2-10 requests per second against a REST API, it is beneficial to download the entire registry and index it manually. In our system, the registry is indexed using Whoosh, a fast, pure Python search engine library. In order to formulate search queries which will return relevant data, it is necessary to identify which part of the transaction

description contains company information and hence be used as search terms in the indexed entity registry. The transaction description is cleaned in the same way as described in Section 2.5 and the first two terms t_1 and t_2 in the resulting string are used to build the query $Q = t_1\ ANDMAYBE\ t_2$.

The ANDMAYBE operator means that we perform the query using t_1 and include t_2 if and only if a match is found while including it. Most of the time the first term describes the transaction well enough to make a successful lookup, but in some cases including the second term may be required. The system is now able to efficiently extract industry codes for transaction texts.

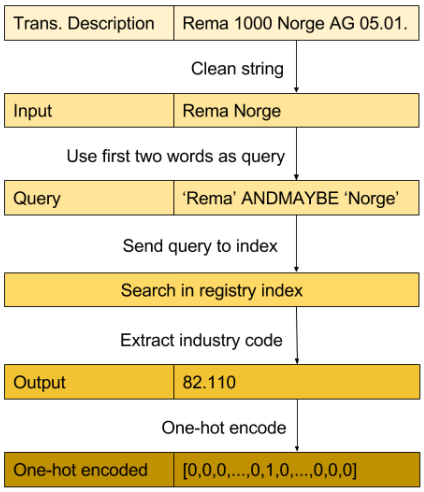

Fig. 5: Industry Code Extraction Example

The industry code uses a representation which is not well suited as input to classification algorithms. It is a 2-part code represented as two numbers divided by a period. The first number represents the industry and the second part specifying the sub-category of said industry. These codes are therefore one-hot encoded and appended to the bag-of-words feature set produced for the baseline (see Fig. 4.d). The transactions for which the system does not find a corresponding entry in the entity registry are assigned a default value of 0 (see Fig. 4.c). This entire process for extracting industry codes is illustrated in Figure 5.

2.7 Google Places API

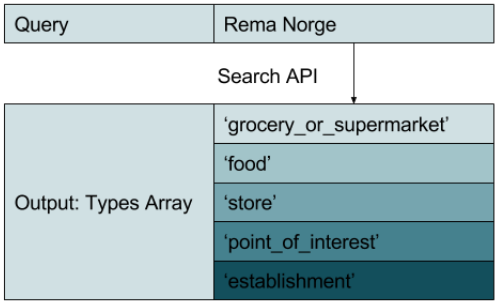

Fig. 6: Google Places API Output Example

The *Google Places API Web Service* is a service that returns information about places — defined within this API as establishments, geographic locations, or prominent points of interest — using HTTP requests[5]. This Web Service allows for a special type of query called Text Search Requests. This request service returns information about a set of places based on a string — for example, "pizza in New York" or "shoe stores near Ottawa" or "123 Main Street"[6]. The service

responds with a list of places matching the text string, each of which contains a number of features. Among these features, there is a feature named 'types,' which is an array of feature types describing the given result.

The types in this array are ordered according to specificity, meaning that the first entry is the most descriptive. An example of a *Google Places* types array is shown in Figure 6. These types are picked from a set of semantically defined types in the *Google Places* API. The first entry is extracted from this array and used as the type describing the transaction. There is likely to be some correlation between this type and the categories representing the transaction texts. It is therefore desirable to extract this data.

Seeing as this data is only accessible through the API and it costs a certain amount per request, it would not be financially or computationally sound to gather this information about every single transaction instance as done with the *Brønnøysund Entity Registry*. Therefore we have chosen a different approach where we identify the subset of transactions which the classifier is not sufficiently confident about and collect *Google Places* data for these transactions only.

In order to identify this subset, the system evaluates the array of distances from the decision boundary of every class that the classifier produces for every transaction. If the distance measurement for a given class is positive, it means that the classifier predicts that the transaction belongs to this class. If it is negative, the classifier predicts it does not belong to the class. So, if there are multiple positive values in this array of distances, the classification model chooses the greatest one, but if there are none, the classification model is saying that the transaction doesn't belong to any of the classes. It is in this last case that we can conclude that the classifier is not sufficiently confident, and the *Google Places* approach is used.

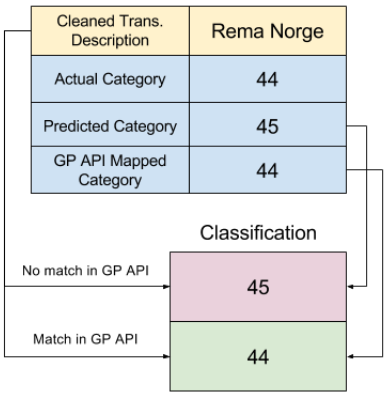

Fig. 7: Google Places API Utilization Example

Of course, we have not trained the classifier on the features gathered from the *Google Places* API so we cannot add them to the feature set to be used as input for the predictor. Therefore a direct mapping between *Google Places* type and transaction categories has been set up. Then, the system looks for a match for all of the non-confident classifications in the *Google Places* API. If there is a match, the mapping between *Google Places* Type and transaction category is used to decide the transaction's class. If there is no match, the system leaves the non-confident classification as it is.

This approach is exemplified in Figure 7 where a transaction with the description "Rema Norge" has been classified by the model to category 45. This classification is deemed non-confident, and a lookup is therefore made in the

Google Places API. If this lookup results in a match, the classification will be changed to the category mapped to by the GP type extracted, which in this case is 44. If the lookup doesn't result in a match, the classification uses the original prediction of category 45. The *Google Places* approach does not handle classification to sub-categories. This is because the types employed in the *Google Places* API are not sufficiently descriptive to be mapped directly to sub-categories.

3 Results

3.1 Experiments

In this section, we describe the basis for which each experiment has been conducted. There is a total of 87199 distinct terms in the transaction texts. We plotted the accuracy of the baseline for Bag-of-Words sizes up to the 20,000 most frequently occurring terms as seen in Figure 8. Here we can see that the accuracy begins to stabilize at size 4,000 making it a reasonable size to use.

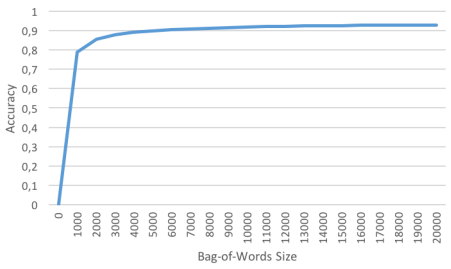

Fig. 8: Accuracy per 1000 increment in Bag-of-Words size

For every experiment the data set is divided into a training and test set, respectively 80% and 20% of the data set. The results given are averages over 100 iterations, shuffling the training and test set each time. In the results obtained from the Baseline and *Brønnøysund Registry* approaches, we may differentiate between *Main Categories* and *Sub Categories*. This means that the target values used for training the model and performing the classifications are the main categories, of which there are 10, or the sub-categories, of which there are 63. In the *Brønnøysund Registry* approach we differentiate between "with" industry code and "exclusively" industry code. "With" means that all transactions are included, and the ones without a match in the registry are given a dummy value of 0 in place of the industry code as shown in Figure 4.c. "Exclusively" means the system uses only the subset of transactions which have a match in the registry and therefore have a corresponding industry code. 192177 (87.13%) of the transactions in the dataset yield a match in the *Brønnøysund Entity Registry* thus constituting the "Exclusive" subset.

The evaluation metrics used are Accuracy (Micro-Averaged Recall), Macro-Averaged Recall, Macro-Averaged Precision and F-Score[2].

3.2 Baseline

In table 3 we observe that the performance measures (recall in particular) are affected by classifying to sub-categories rather than main categories.

Table 3: Evaluation scores for the baseline

Target Categories	Accuracy	Recall	Precision	F-Score
Main Categories	0,8922	0,8668	0,9322	0,8951
Sub Categories	0,8632	**0,7048**	0,8934	0,7707

Table 4: Baseline Per Class Results. Shows the evaluation scores of each class.

Main Category	Precision	Recall	F-Score
42	0.96	0.88	0.92
43	0.94	0.87	0.90
44	0.98	0.92	0.95
45	0.76	0.96	0.85
47	0.88	0.81	0.85
48	0.93	0.74	0.83
49	0.93	0.83	0.88
103	0.96	0.81	0.88
104	0.99	0.88	0.93
181	0.99	0.98	0.98

3.3 *Brønnøysund Entity Registry*

In Table 5 we observe that there is a slight gap in accuracy between the exclusive and non-exclusive transaction sets. We also see that the performance measures (recall in particular) are affected by classifying to sub-categories rather than main categories.

Table 5: Brønnøysund Registry Results. Shows the model's evaluation scores after the industry codes from the *Brønnøysund Registry* have been added to the feature set.

Target	Brreg	Accuracy	Recall	Precision	F-Score
Main Cat.	Exclusively	**0,9380**	0,9226	0,9466	0,9338
Main Cat.	With	**0,9201**	0,8993	0,9395	0,9177
Sub Cat.	Exclusively	0,9192	**0,7936**	0,8825	0,8253
Sub Cat.	With	0,8918	**0,7559**	0,8764	0,8011

Table 6: Brønnøysund Registry Per Class Results. shows the evaluation results of each class using the *Brønnøysund Registry* approach.

Main Category	Precision	Recall	F-Score
42	0.96	0.92	0.94
43	0.93	0.93	0.93
44	0.97	0.94	0.95
45	0.87	0.97	0.92
47	0.94	0.91	0.93
48	0.93	0.80	0.86
49	0.94	0.91	0.92
103	0.93	0.84	0.88
104	0.98	0.92	0.95
181	0.98	0.99	0.99

3.4 *Google Places* API

Table 7: Google Places Key Metrics. Count refers to the share of non-confident classifications. API Matches is the percentage of non-confident classifications which yield a match in GP API. Positive is the share of API matches which map to correct class. The two last columns refer to the share of API matches which lead to positive and negative alterations of the classification.

Count	API Matches	Positive	False -> Positive	Positive -> False
13.94%	65.60%	43.99%	23.68%	1.98%

Table 8: Google Places Results. Shows the evaluation scores for the model after implementing the *Google Places* approach.

Accuracy	Recall	Precision	F-Score
0.9123	0.8886	0.9369	0.9100

Table 9: Google Places Per Class Results. Shows the evaluation results of each class using the *Google Places* approach.

Main Category	Precision	Recall	F-Score
42	0.97	0.90	0.93
43	0.94	0.90	0.92
44	0.97	0.93	0.95
45	0.81	0.97	0.89
47	0.92	0.88	0.90
48	0.93	0.74	0.83
49	0.94	0.87	0.90
103	0.94	0.83	0.88
104	0.98	0.91	0.94
181	0.99	0.98	0.98

Every class is affected by positive and negative classification changes. If we normalize the percentages of negative and positive classification changes for each class by taking those values and multiplying them by their corresponding weights in Table 7, respectively 0.2368 and 0.0198 for negative and positive classification changes, we get a measure of how much each class is affected by the classification changes. Calculating the difference between these yields a value which indicates whether or not the approach contributes positively (> 0) or negatively (< 0) towards the accuracy in a given class. This is shown in Table 10.

Table 10: Per Class Classification Change Contribution

Norm. Positive Class Change	Norm. Negative Class Change	Class contribution (Diff.)
3,50	0,24	3,26
4,63	0,15	4,48
0,35	0,24	0,11
3,16	0,67	2,50
6,22	0,25	6,00
0,95	0,14	0,81
4,13	0,26	3,87
0,15	0,02	0,13
0,31	0	0,31
0,27	0,03	0,24

3.5 Combining Approaches

Table 11: Combined Approaches results. Shows the evaluation scores for the classification model when applying both the *Google Places* and the *Brønnøysund Entity Registry* approaches.

Accuracy	Recall	Precision	F-Score
0,9297	0,9088	0,9426	0,9243

Table 12: Combined Approaches Per Class Results. Shows the evaluation results of each class using a combination of the *Google* Places and *Brønnøysund Registry* approaches.

Main Category	Precision	Recall	F-Score
42	0.96	0.92	0.94
43	0.93	0.93	0.93
44	0.97	0.94	0.95
45	0.87	0.97	0.92
47	0.94	0.91	0.93
48	0.93	0.80	0.86
49	0.94	0.91	0.92
103	0.93	0.84	0.88
104	0.98	0.92	0.95
181	0.98	0.99	0.99

4 Discussion

To open this section, we will briefly introduce what we identify as the most important topics of discussion. We will discuss the results produced with regards to the *Brønnøysund Registry* and *Google Places* approaches both individually and combined. We will explain what their significance is and discuss what implications they may have. By doing this, we hope to shed light on what we believe to be the reasons behind why the results turned out as they did.

Table 13: Percentage point improvements. Shows the improvement in evaluation scores each of the approaches made in relation to the Baseline.

Approach	Accuracy	Recall	Precision	F-Score
Brønnøysund Registry	**2,79 %**	**3,25 %**	0,73 %	2,26 %
Google Places	**2,01 %**	**2,18 %**	0,47 %	1,49 %
Combination	**3,75 %**	**4,20 %**	1,04 %	2,92 %

4.1 *The Brønnøysund Registry*

The intuition behind utilizing the industry codes extracted from the *Brønnøysund Entity Registry* was that they would be somewhat correlated to the target values for our transactions. This led to the hypothesis that using them to extend our feature set would lead to an increase in the accuracy of our classification model. Our results show an increase in accuracy of 4.58 and 2.79 percentage points respectively for the exclusive and non-exclusive methods of evaluating the approach. Exclusive here referring to testing on the subset of our data for which we were able to extract industry codes.

The gap in accuracy between the exclusive and non-exclusive evaluations may have occurred for two possible reasons. The first is that the exclusive subset has a distribution of transactions which are more easily classified. The second reason could be that when using the exclusive subset, the classifier is not affected by the 'dummy' value which is assigned to all transactions without a corresponding industry code. When we assign this 'dummy'-value, we are telling the classifier that all the transactions with this value have something in common, when in reality they may have nothing in common.

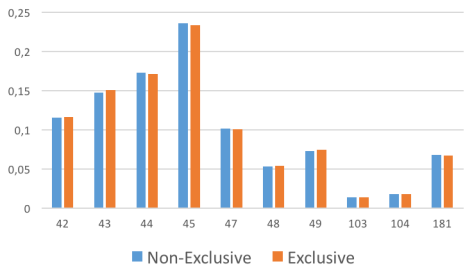

Fig. 9: Comparison of label distributions for data set and subset with industry codes

In order to identify which of these two reasons contribute to the gap in accuracy, we look at the label distributions for the exclusive and non-exclusive

transaction sets shown in Figure 9. These are approximately the same, indicating that the baseline results should be approximately the same in both cases. However, if we compare the per class results for the *Brønnøysund Registry* approach in Table 6 and the Baseline in Table 4, we see that the former performs better for the larger classes (43, 44, and 45). This could explain the gap in accuracy since the transactions without industry codes are not diminishing the effects of the *Brønnøysund Registry* approach in the exclusive subset. In other words, this indicates that replacing missing industry codes with a 'dummy'-value is the factor which causes this accuracy gap between the exclusive and non-exclusive transaction sets.

The ideal situation would be to have industry codes for all transactions, but we are only able to retrieve industry codes for approximately 87% of all transactions. We, therefore, decided to use the 'dummy'-values and accept the loss in contributed accuracy from the *Brønnøysund Registry* approach.

The *Brønnøysund Registry* approach adds very little overhead to the running time of the system. This is because it has been downloaded and indexed, and therefore can be queried locally. The downside to this approach is that the index is not kept up to date automatically. As we can see in both the Baseline and *Brønnøysund Registry* results, the evaluation scores fall significantly when classifying to the sub-categories. This is because the complexity of separating the data increases with the number classes.

4.2 *Google Places* API

This approach is a post-processing technique which aims to identify classifications which are believed to be incorrect and attempt to reclassify them to increase the accuracy of the system. The approach identifies 13.94% of the classifications as non-confident. These are the classifications which the system will try to reclassify by searching for a match in the *Google Places* API. Of these classification instances, we are able to find a match in the GP API for 65.6% of them, and 43.99% of these result in a correct classification. This means that as a stand-alone classifier it would achieve an accuracy score of approximately 28% (product of the number of matches and number of correct classifications), which is very poor.

If there is a match for a given transaction in the *Google Places* API, this approach can have four outcomes:

- False -> Positive: GP mapping changes incorrect prediction to correct.
- False -> False: GP mapping changes incorrect prediction to same or other incorrect prediction.
- Positive -> Positive: GP mapping leaves prediction unchanged.
- Positive -> False: GP mapping changes correct prediction to incorrect.

We refer to these outcomes as classification changes. The Positive-to-Positive and False-to-False classification changes are not interesting as they will have no effect on the accuracy of the system. It is desirable to maximize the False-to-Positive classification changes as these will increase accuracy, and minimize

Positive-to False-classification changes as these will decrease accuracy. As we can see in Table 10 the class contributions are positive for all classes meaning that positive classification changes outnumber the negative classification changes in all classes. If this were not the case, we could omit certain classes from the *Google Places* approach in order to increase its efficiency.

Ultimately, the *Google Places* approach leads to a 2.01 percentage point increase in accuracy compared to the baseline. It is, however, a time-consuming procedure as we are required to make requests to a REST API for all non-confident classifications.

4.3 Combining Approaches

When we combine the two approaches discussed in this paper, we would expect to reap the benefits of both approaches. This is almost the case, but there is a slight overlap between the two approaches when it comes to which transactions they improve the accuracy for. In the classes where there is no overlap, the contribution in accuracy from the two approaches separately should equal the contribution of the approaches in combination. If the combined contribution is smaller than the sum of individual contributions, then there is an overlap in the transactions they correctly classify.

If we look at table 14 we can see the difference between combined contribution and sum of individual contributions defined as the overlap measure. If the overlap measure is 0, there is no overlap, if it is negative its magnitude determines the amount of overlap in the class. We observe that six of the ten of the classes are affected by this overlap.

Table 14: Per class overlap measure between approaches. The second column shows the sum of the improvements contributed by the two approaches individually. The third column shows the improvement contributed by the approaches in combination. The final column shows the overlap measure.

Main Category	**Sum indiv. approach**	**Combined approach**	**Overlap Measure**
42	0,04	0,05	-0,01
43	0,06	0,08	-0,02
44	0,02	0,02	0
45	0,01	0,01	0
47	0,1	0,14	-0,04
48	0,06	0,06	0
49	0,08	0,09	-0,01
103	0,03	0,05	-0,02
104	0,04	0,06	-0,02
181	0,01	0,01	0

Our combined approach yielded an accuracy of 92.97%, and seeing as our human classifier experiment resulted in an average accuracy of 93% we can argue that our data does not provide enough information for classification methods to achieve evaluation scores that are much higher than this.

5 Related Work

A project conducted by Skeppe[3] attempts to improve on an already automatic process of classification of transactions using machine learning. No significant improvements were made using fusion of transaction information in either early or late fusion. The results do however show that bank transactions are well suited for machine learning, and that linear supervised approaches can yield acceptable scores.

In Gutiérrez et al.[8] they use an external semantic resource to supplement sentences designated for sentiment classification. The resource and methods they propose reach the level of state-of-the-art approaches.

In the study conducted by Albitar[9], classification of text is performed using a Bag-of-Words Model which is conceptualized and turned into a Bag-of-Concepts Model. This model is then enriched using related concepts extracted from external semantic resources. Two semantic enrichment strategies are employed, the first one is based on a semantic kernel method while the second one is based on a method of enriching vectors. Only the second strategy reported better results than those obtained without enrichment.

Iftene et al.[10] present a system designed to perform diversification in an image retrieval system, using semantic resources like YAGO, Wikipedia, and WordNet, in order to increase hit rates and relevance when matching text searches to image tags. Their results show an improvement in terms of relevance when there is more than one concept in the same query.

In the research conducted by Ye et al.[11] a novel feature space enriching (FSE) technique to address the problem of sparse and noisy feature space in email classification. The FSE technique employs two semantic knowledge bases to enrich the original sparse feature space. Experiments on an enterprise email dataset have shown that the FSE technique is effective for improving the email classification performance.

Poyraz et al.[12] perform an empirical analysis the effect of using Turkish Wikipedia (Vikipedi) as a semantic resource in the classification of Turkish documents. Their results demonstrate that the performance of classification algorithms can be improved by exploiting Vikipedi concepts. Additionally, they show that Vikipedi concepts have surprisingly large coverage in their datasets which mostly consist of Turkish newspaper articles.

In our research, we have combined feature enrichment using external semantic resources with the classification of real bank transactions. This is an important intersection that needs further research. We hope to have laid a foundation upon which others can continue research in the domain of classification of financial data.

6 Conclusion

Our results show that using external semantic resources to supplement the classification model provides a significant improvement to the overall accuracy of the system. The *Brønnøysund Registry* approach has proven to be the best contributor, both regarding the increase in accuracy, and the low running time as it requires minimal overhead compared to the *Google Places* approach. These approaches can be directly translated to other external semantic resources and therefore provide a robust method of extending classification models.

In order to further increase the accuracy of the system, we would propose to explore which other external resources could be used in combination with the approaches described in this project. We would also recommend exploring other representations than Bag-of-Words to see if this could have a positive impact on the accuracy of the system. A multi-label classification solution for this data could also be a potentially useful area to study.

References

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics)", Springer-Verlag New York, Inc., Secaucus, NJ, (2006).
2. Van Asch, Vincent. "Macro-and micro-averaged evaluation measures." Available at `https://www.semanticscholar.org/` (2013). Last accessed 23/4/2017.
3. Lovisa B. Skeppe, "Classifying Swedish Bank Transactions with Early and Late Fusion Techniques." Master Thesis. KTH Royal Institute of Technology, Stockholm (2014).
4. Claudia Perlich, "Which is your favourite Machine Learning Algorithm?" Available at `http://www.kdnuggets.com/2016/09/perlich-favorite-machine-learning-algorithm.html` (2016). Last accessed 10/5/2017.
5. The Google Places API Web Service. Available at `https://developers.google.com/places/web-service/intro`. Last accessed 15/6/2017.
6. The Google Places API Text Search Requests. Available at `https://developers.google.com/places/web-service/search#TextSearchRequests`. Last accessed 15/6/2017.
7. Erlend Vollset, Eirik Folkestad, "Automatic Classification of Bank Transactions," Chapter 2. Master Thesis. Norwegian University of Science and Technology, Trondheim (2017).
8. Yoan Gutiérrez, Sonia Vázquez, Andrés Montoyo, "Sentiment classification using semantic features extracted from WordNet-based resources," Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, p.139-145 (2011)
9. Albitar, S., Espinasse, B., Fournier, S., "Semantic Enrichments in Text Supervised Classification: Application to Medical Domain," Florida Artificial Intelligence Research Society Conference, (2014).
10. Iftene, A., Baboi, A.M. "Using semantic resources in image retrieval." 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, Vol. 96, Pp. 436-445, Elsevier (2016).
11. Ye Y., Ma F., Rong H., Huang J.Z. "Improved Email Classification through Enriched Feature Space." In: Li Q., Wang G., Feng L. (eds) Advances in Web-Age Information Management (WAIM), (2004).
12. Poyraz, M., Ganiz, M. C., Akyokus, S., Gorener, B., and Kilimci, Z. H. "Exploiting Turkish Wikipedia as a semantic resource for text classification." International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 1-5 (2013).

Why Enriching Business Transactions with Linked Open Data May be Problematic in Classification Tasks

Eirik Folkestad
Department of Computer Science
Norwegian University of Science and Technology

Erlend Vollset
Department of Computer Science
Norwegian University of Science and Technology

Marius Rise Gallala
Sparebank1 SMN

Jon Atle Gulla
Department of Computer Science
Norwegian University of Science and Technology

***Abstract*—Linked Open Data has proven useful in disambiguation and query extension tasks, but their incomplete and inconsistent nature may make them less useful in analyzing brief, low-level business transactions. In this paper, we investigate the effect of using Wikidata and DBpedia to aid in classification of real bank transactions. The experiments indicate that Linked Open Data may have the potential to supplement transaction classification systems effectively. However, given the nature of the transaction data used in this research and the current state of Wikidata and DBpedia, the extracted data has in fact a negative impact the accuracy on the classification model when compared to the Baseline approach. The Baseline approach produces an accuracy score of 88,60% where the Wikidata, DBpedia and their combined approaches yield accuracy scores of 84,99%, 86,65% and 83,48%.**

Keywords*—*Classification, Bank Transactions, Logistic Regression, External Data, Linked Open Data, Wikidata, DBpedia.

I. Introduction

This project is carried out in cooperation with Sparebank1, which is Norway's largest regional bank and the second largest Norwegian-owned bank, to gain insight into classification of real bank transactions. Progress in the domain at the intersection of finance and machine learning is important as it has a lot of potential applications like accurate consumption statistics, financial trend predictions, or financial crime detection to name a few. In the research that we have previously conducted on automatic classification of bank transactions [12], an approach that utilizes transaction description texts to classify transactions has been developed. This approach has proven to be somewhat effective with a classification accuracy of 88,6%, but we wish to develop techniques to further improve this approach.

Linked open data can be difficult to apply in domains that require a high level of data consistency. This is because linked open data sources contain large volumes of data but the quality varies a great deal. The research in this paper aims to investigate methods for exploiting linked open data to aid in classification of business transactions and identify why this can be a problematic task. We examine two linked open resources; *Wikidata*, which is a collaboratively edited knowledge base operated by the Wikimedia Foundation and is intended to provide a common source of data, and *DBpedia*, which allows users to semantically query relationships and properties of Wikipedia resources, including links to other related datasets. Two main approaches to the problem are covered:

- Using extracted linked open data to extend the baseline feature set
- Enhancing original and extracted data to better exploit the linked open data

This paper gives a detailed description of the implementation of two approaches which do not lead to improvement of performance. We do, however, provide a thorough analysis of the results obtained from testing the system, giving valuable insight into why the use of Linked Open Data as a semantic feature enrichment tool is a difficult task. Due to the general nature of the techniques in this project, they can, without too much trouble, be transferred to other linked open data sources and other applications within text classification.

The remainder of this paper is structured as follows. In Section II we present studies that are related to the conducted work in this paper. Section III describes the theoretical foundation upon which we have built our project as well as giving a detailed description of the data we have used and to what extent it is represented. Section IV follows with providing a presentation of the experiments we have conducted, as well as the results we obtained from them. The project is summarized in sections V and VI by discussing our findings, providing recommendations for further work, and drawing our final conclusions.

II. Related Work

A project conducted by Skeppe[10] attempts to improve on an already automatic process of classification of transactions using machine learning. No significant improvements were made using a fusion of transaction information in either early or late fusion. The results do however show that bank transactions are well suited for machine learning, and that linear supervised approaches can yield acceptable scores.

In Gutiérrez et al.[13] they use an external semantic resource to supplement sentences designated for sentiment classification. The resource and methods they propose reach the level of state-of-the-art approaches. In the study conducted by Albitar[14], classification of text is performed using a Bag-of-Words Model which is conceptualized and turned into a Bag-of-Concepts Model. This model is then enriched using

related concepts extracted from external semantic resources. Two semantic enrichment strategies are employed, the first one is based on semantic kernel method while the second one is based on enriching vectors method. Only the second strategy reported better results than those obtained without enrichment.

Iftene et al.[15] present a system designed to perform diversification in an image retrieval system, using semantic resources like YAGO, Wikipedia, and WordNet, to increase hit rates and relevance when matching text searches to image tags. Their results show an improvement regarding relevance when there is more than one concept in the same query.

In the research conducted by Ye et al.[16] a novel feature space enriching (FSE) technique to address the problem of sparse and noisy feature space in email classification. The (FSE) technique employs two semantic knowledge bases to enrich the original sparse feature space. Experiments on an enterprise email dataset have shown that the FSE technique is effective for improving the email classification performance.

Poyraz et al.[17] perform an empirical analysis the effect of using Turkish Wikipedia (Vikipedi) as a semantic resource in the classification of Turkish documents. Their results demonstrate that the performance of classification algorithms can be improved by exploiting Vikipedi concepts. Additionally, they show that Vikipedi concepts have surprisingly large coverage in their datasets which mostly consist of Turkish newspaper articles.

Xiong et al.[7] present a simple and effective method of using a knowledge base, Freebase, to improve query expansion, a classic and widely studied information retrieval task. By using a supervised model to combine information derived from Freebase descriptions and categories to select terms that are useful for query expansion. Experiments done on the ClueWeb09 dataset with TREC Web Track queries demonstrate that these methods are almost 30% more successful than strong, state-of-the-art query expansion algorithms. Some of these methods also have 50% fewer queries damaged which yield better win/loss ratios than baseline algorithms.

In our research, we have combined feature enrichment using external linked open data resources with classification of real bank transactions. This is an important intersection that needs further research. We hope to have laid a foundation upon which others can continue research in the domain of classification of financial data.

III. Data and Methods

In the following section, the author describes in detail the techniques which have been implemented and the data they have been used on. The theoretical foundation of this project is explained in detail in the appendix.

A. *Original Data set*

The original data set used in this project consists of 220618 records of unstructured real Norwegian bank transaction texts from Sparebank1. The transaction texts have a corresponding category (C) and sub-category (SC) of which the transaction belong in. There is a total of 10 main categories and 63 sub-categories. In this project, we will only do experiments with the main categories which make for a good indication of the performance of our selected approaches. The main categories are shown in table I and an example of a transaction text can be seen in Table II.

Main Category ID	Main Category Name	Main Category Name in English
42	Bil og transport	Automobile and Transport
43	Bolig og eiendom	Housing and Real-Estate
44	Dagligvarer	Groceries
45	Opplevelse og fritid	Recreation and Leisure
47	Helse og velvære	Health and Well Being
48	Hobby og kunnskap	Hobby and Knowledge
49	Klær og utstyr	Clothes and Equipment
103	Annet	Other
104	Kontanter og kredittkort	Cash and Credit
181	Finansielle tjenester	Financial Services

TABLE I: Main Categories and their IDs

Description	SC	C
Rema 1000 Norge AG 05.01.	61	44

TABLE II: Data entry example

By conducting a Simple Random Sample of 400 transactions transactions and manually labeling them by the main category, we achieved an accuracy of 93%. The mislabeling was mostly due to poor quality of the data where business names were not present in the text. Accuracy scores which are close to 93% is therefore considered to be great accuracy scores.

B. *Bag-of-Words Model*

The Bag-of-Words (BoW) model is used to convert text to a representation which is better suited for many machine learning algorithms. This technique is commonly used in natural language processing and information retrieval. In our application of the model, it is used as a tool for feature generation from the bank transaction texts. Each text is represented as a multiset (bag) of terms contained in the text when generating features for a corpus of texts. Given a corpus of the texts $X = x_1, x_2$, where

x_1 = Greg has a table

x_2 = A table is a table

the BoW representation produced is shown in table III. The resulting matrix has a row for each text and a column for each term in the corpus. The value of a cell is the frequency of the represented term in a given text. The features can then be used as input to a predictive model such as the one in this project.

X	Alan	has	a	chair	is
x_1	1	1	1	1	0
x_2	0	0	2	2	1

TABLE III: Bag-of-words example

C. *Logistic Regression*

The Logistic regression classification algorithm is linear and estimates a probability of a class Y given a feature-vector X. It does this by using a logistic function to find

the relationship between the class and the feature-vector. It assumes that the distribution P(Y|X), where Y is the class and X is the feature-vector, is on a parametric form and then estimates it from the training data. The probability P(Y|X) of X belonging to class Y is given by the sigmoidal function which we can see in Eq. 1 and Eq. 2.

$$z(Y, X) = \sum_{i=1}^{N} w_i f_i(Y, X) \tag{1}$$

$$P(Y|X) = \frac{1}{1 + exp(-z(Y, X))} \tag{2}$$

P(Y|X) is estimated by linearly combining the features of X multiplied by some weight w_i and applying a function $f_i(Y, X)$ on the combinations. f_i is a function which returns a relationship value between a feature of a class and a feature in a feature-vector in the form of true or false based on the probability being over a certain threshold. Some features are more important than others, so the weight w_i denotes the "strength" of the feature.

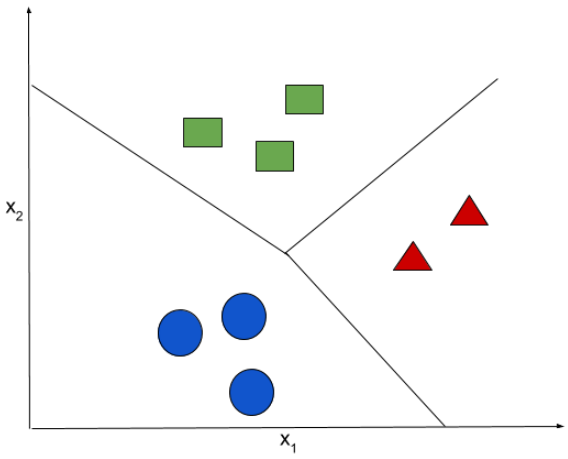

Fig. 1: Softmax Regression example

In this project, we need a classifier which can handle multiple classes. A variant of a Logistic Regression classifier which can handle more than two classes is the Softmax Regression classifier. In Softmax Regression, we replace the logistic function in Eq. 2 with the SoftMax function as we see in Eq. 3 which gives the probability of each class [8].

$$P(y|X)_{y \in Y} = \frac{exp(z(y, X))}{\sum_{y' \in Y} exp(z(y', X))} \tag{3}$$

From the expression in 3 it can be shown that $\sum_{y \in Y} P(y|X) = 1$. This leads to the following classifier in 4 for a feature-vector X which outputs the class $\hat{y}$ if only the class of the feature-vector is needed and not the probability itself. An example of a Softmax Regression classifier can be seen in Fig. 1.

$$\hat{y} = argmax_{y \in Y} P(y|X) \tag{4}$$

D. Baseline

A baseline refers to a set of configurations and techniques applied to our system used as a basis for defining change measuring improvement. In our system, the baseline approach is a standard machine learning approach to text classification which involves using a Bag-of-Words representation and Softmax Regression for classification.

1) Preprocessing: Before the text can be used in a classification algorithm, some preprocessing steps are applied to the data. To remove noise from the data, we clean the description string to remove all punctuation, numbers, and words shorter than three letters (see Fig. 4b). The text is then transformed to vector representation using the bag-of-words model (see Fig. 4c) and is ready for use in a classification algorithm.

2) Classification: The classification algorithm we apply in the baseline approach is Logistic Regression using a Softmax scheme which creates a true multinomial classifier of which can be used to classify data based the highest probability yielded of the likelihood of belonging to one of the multiple classes. The choice of a classifier is based on finding out that the Logistic Regression produces promising results where a more complex classifier like a Feed-Forward Neural Network does not improve the results [11]. From this, we can conclude that the data used in this project is linearly separable and a simpler classifier like a Softmax Regression classifier will be sufficient.

E. Wikidata and DBpedia

Wikidata and DBpedia are both Linked Open Data knowledge bases for extracting structured data from the web. Wikidata is a user-curated source of structured information which is included in Wikipedia and DBpedia provides structured data from the Wikipedia and Wikimedia Commons [1].

Both linked open data sources are structured in a hierarchy consisting of objects where their hierarchical relationships are described with RDF-triples. A RDF-triple contains three components; subject, predicate and object [6]. An example of an RDF-triple in Wikidata or DBpedia related to the project can be seen in Fig. 2.

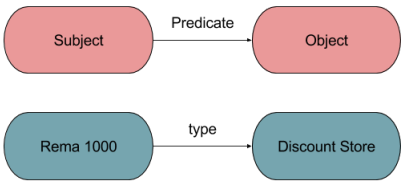

Fig. 2: RDF-Triple Structure and Example

We want to acquire the meaning of the words in the transaction texts with the use of Wikidata and DBpedia. A visible trend in the original data is that a company name usually is present in the transaction texts so an approach would be to find the company name with the help of Wikidata and DBpedia, and find information about what industry the company operates in.

One API-call per transaction text would be very time consuming since our system can only do few API calls per second and the original data set is of size 220618. Since both Wikidata and DBpedia support queries through a SPARQL-endpoint that is capable of returning thousands of results and we are looking for something specific, a less time-consuming approach would be to find all companies and a description of what they do that Wikidata and DBpedia have structured data for and store it to a local file. We also do not need all the information that Wikidata and DBpedia can offer us about each company. An assumption of what would benefit training a prediction model the most would be a short description which specifically states something about what industry the company operates in. The closest predicate of which we could find that would fit our needs in Wikidata was *Description* and *Subjects* in DBpedia. The *Industry* predicate was considered and seemed more promising than *Subjects* in DBpedia but relatively few companies used this predicate, unfortunately. The query results for Wikidata and DBpedia were stored in separate local files, and the two were indexed to separate indexes with the company name as key and the description as value by using *Whoosh* [4] for quick and reliant look-up.

From the companies in our index, we can find useful information about companies in the transaction texts as seen in Fig. 3. First, the transaction text is cleaned to remove all punctuation, numbers, and words shorter than three letters. Then we search our index for the first and/or the second word in the transaction text which represent the company name and if a result is returned, we extend our transaction text. After this the process of Fig. 4 is applied, and the new transaction text is converted to a Bag-of-Words representation. Depending on the information used to extend the original data is from the Wikidata index or the DBpedia index, the approaches are called the Wikidata approach or the DBpedia approach. If the information is extended from both Wikidata and DBpedia the approach is called the Wikidata & DBpedia approach.

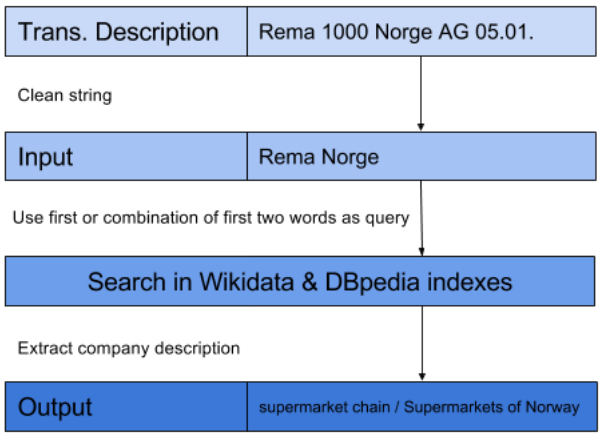

Fig. 3: Wikidata and DBpedia Description Extraciton Example

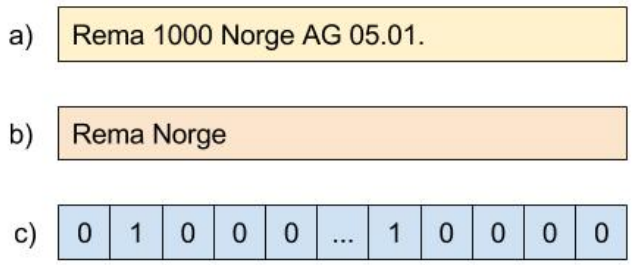

Fig. 4: Transaction Representation Example
(a) Transaction Description
(b) Transaction Description Cleaned
(c) Bag-of-Words representation

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

F. *WordNet*

WordNet is a large lexical database of the English language and can be used for searching for definitions, synonyms and other information about English words [2]. It can also be used for simple translation from a supported language to English before doing a search. The information about the word will be returned in English.

Natural Language Toolkit [3] provides a module that can be downloaded so that WordNet is available locally. This means that no calls to an API are needed. This will make the process of searching for information about words much faster.

By using the WordNet module that *Natural Language Toolkit* provides, a word can be sent in, and synonyms are returned if there are any (see Fig. 5).

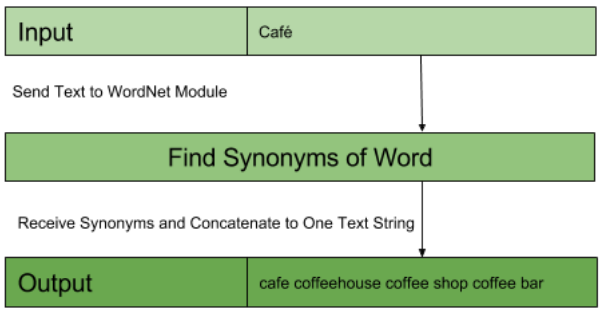

Fig. 5: WordNet - Extraction of Synonyms for a Word

We are trying to bring even more semantic meaning into the transaction texts by using synonyms. Often the same meaning is represented by using words that are synonyms. For instance, a café can be represented by the word coffee shop. By adding synonyms to a text, we can create similarities between two texts that are initially viewed as dissimilarities since the words are written differently.

With the help of WordNet we intend to extend the transaction texts, and also the descriptions we receive from Wikidata

and DBpedia, with synonyms so that similarities between two or more records that originally are not represented will be more transparent as they now share more words. As seen in (see Fig. 6) the data is first cleaned by removing punctuation, numbers, stopwords, and words that are shorter than three and then split to get each word separate. Each word is sent to the WordNet module, and the synonyms are returned. The words are then concatenated to one text string again. Data which i.e. contain the word 'bar' would now share this word with a text which contains the word 'cafe'. The general similarities between two texts are now clearer after extending the data with synonyms. After extending the transaction text with synonyms the process in Fig. 4 is applied, and the new transaction text is converted to a Bag-of-Words representation.

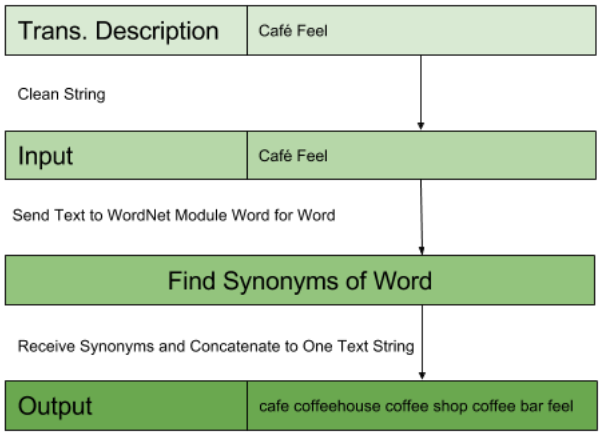

Fig. 6: WordNet - Extraction of Synonyms for a Transaction

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

G. *Yandex Translation*

Yandex is a technology company that builds intelligent products and services powered by machine learning [5] and one of these services is a translation API that seems fit to translate the transaction texts in the original data set.

The transaction texts used in this project are in Norwegian, and this could create problems when using the selected linked open data sources which returned descriptions are in English. By translating the original data to English, we hope to compensate for the possible problems created by extending data with data on a different language. The translated transaction texts will hopefully share more words with the descriptions from the linked open data sources.

The Yandex Translate module can translate the original data word for word instead of translating the whole texts. This is done since it can be unfavorable to change the idiomatic meaning of the text and rather replace the individual words with their translations.

As shown in Fig. 7 the translation is extracted by first cleaning the text by removing punctuation, stopwords, numbers, and words shorter than three characters and then splitting the texts into separate words. We then translate each word respectively with the translation API. The returned translation of each word is then concatenated into one text string which constitutes the new transaction text. The process of Fig. 4 is then applied, and the translated transaction texts are then converted to a Bag-of-Words representation.

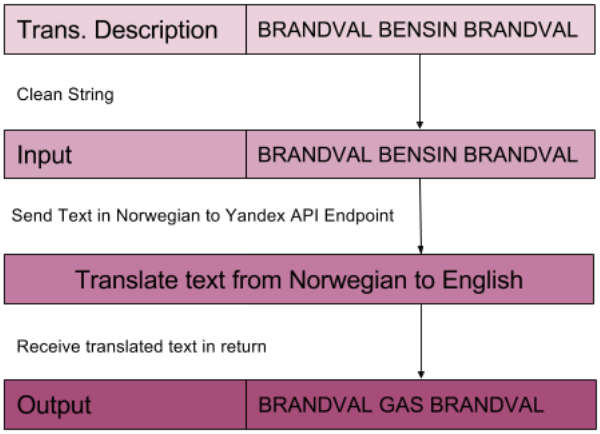

Fig. 7: Yandex - Extraction of Translation for a Transaction

After the new transaction text is represented with a Bag-of-Words Model, we can apply the Softmax Regression classification algorithm to solve our classification problem.

IV. Results

A. *Experiments*

All experiments have been conducted with the following parameters:

- Bag-of-Words size of 4000.
- Logistic Regression with the Softmax function which enables a multinomial classification model, also known as Softmax Regression.
- Classification classes are the 10 Main Categories.

There is a total of 87199 distinct terms in the transaction texts. Using a Bag-of-Words size of 4000 means using the 4000 most frequently occurring terms. We plotted the accuracy of the baseline for Bag-of-Words sizes up to 20,000 as seen in Figure 8. Here we can see that the accuracy begins to stabilize at size 4000 thus making it a reasonable size to use.

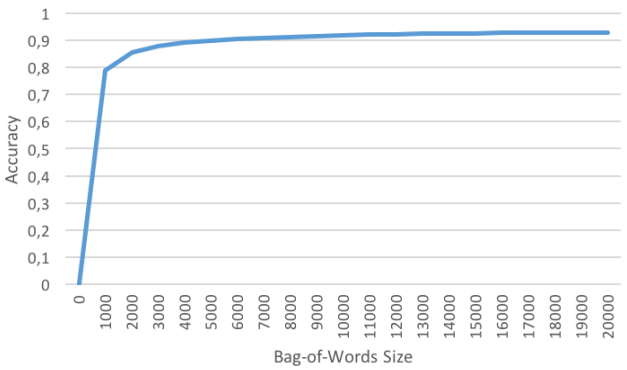

Fig. 8: Accuracy per 1000 increment in Bag-of-Words size

In our experiments we use a number of subsets of the **Original Data Set (ODS)**, which is defined in Sec. III-A. Since the sizes of these subsets are significantly smaller than ODS, we would also expect the results to be different. These subsets are defined as:

- **Wikidata Exclusive Subset (WES)** - This subset consists of only the bank transactions which yield a match in the Wikidata data source. The subset contains **113263** transactions and is used only in the Baseline, and Wikidata approaches.
- **DBpedia Exclusive Subset (DES)** - This subset consists of only the bank transactions which yield a match in the DBpedia data source. The subset contains **125765** transactions and is used only in the Baseline and DBpedia approaches.
- **Wikidata & DBpedia Exclusive Subset (WDES)** - This subset is the union of the Wikidata Exclusive Subset and the DBpedia Exclusive Data Set. The subset contains **136474** transactions and is used only in the Baseline, and Wikidata & DBpedia approaches.

For every experiment the data set is divided into a training and test set, respectively 80% and 20% of the data set. The results given are averages of performance measures over ten iterations, shuffling the training and test set each time.

The label parameters of the tables explained:

- **Approach** - Represent which approach or approaches is used which can involve extending data and enhancing data. The use of the **Translation** and **Synonyms** approaches is to aid in the use of the linked open data approaches.
- **Category** - Represent the Category ID number.
- **Data Set** - Represent which data set is used.
- **Accuracy** - Represent the Accuracy measure score in percent.
- **Recall** - Represent the Macro-Averaged Recall measure score in percent [9].
- **Precision** - Represent the Macro-Averaged Precision measure score in percent [9].
- **F-Score** - Represent the F-Score measure score in percent.

B. *Baseline*

Table IV shows the model's evaluation scores of the Baseline on a Bag-of-Words representation using various data sets.

Approach	Data Set	Accuracy	Recall	Precision	F-Score
Baseline	ODS	88.60%	92.80%	85.35%	88.53%
Baseline	WES	94.39%	94.13%	90.57%	92.17%
Baseline	DES	94.26%	94.65%	90.67%	92.46%
Baseline	WDES	94.08%	94.40%	90.70%	92.39%

TABLE IV: Baseline Approach Results on Various Data Sets

The Table V shows the performance in each Main class.

Main Category	Precision	Recall	F-Score
42	0.91	0.94	0.92
43	0.91	0.92	0.92
44	0.94	0.97	0.95
45	0.94	0.84	0.89
47	0.90	0.90	0.90
48	0.78	0.90	0.84
49	0.86	0.93	0.89
103	0.84	0.93	0.88
104	0.88	0.96	0.92
181	0.97	0.96	0.97

TABLE V: Per Class Results for the Baseline Approach

C. *Use of Linked Open Data*

The approaches in this section are extended and/or enhanced variants of the baseline (see Sec. IV-B) which means that the original data have been altered or appended to.

*1) **Wikidata**:* Table VI shows the model's evaluation scores after the descriptions from Wikidata have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Wikidata	84.65%	89.85%	81.56%	84.92%
Wikidata + Translation	84.99%	88.97%	82.01%	84.90%
Wikidata + Translation + Synonyms	79.87%	85.27%	76.23%	79.79%

TABLE VI: Wikidata Approach Results on the Original Data Set

Table VII shows the model's evaluation scores after the descriptions from Wikidata have been used to extend the data in the Wikidata Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Wikidata	92.69%	92.19%	87.89%	89.77%
Wikidata + Translation	93.15%	92.11%	88.33%	89.98%
Wikidata + Translation + Synonyms	91.83%	90.96%	87.04%	88.72%

TABLE VII: Wikidata Approach Results on the Wikidata Exclusive Subset

*2) **DBpedia**:* Table VIII shows the model's evaluation scores after the descriptions from DBpedia have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
DBpedia	86.48%	90.65%	83.74%	86.58%
DBpedia + Translation	86.65%	89.85%	84.20%	86.59%
DBpedia + Translation + Synonyms	81.74%	86.19%	78.33%	81.46%

TABLE VIII: DBpedia Approach Results on the Original Data Set

Table IX shows the model's evaluation scores after the descriptions from DBpedia have been used to extend the data in the DBpedia Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
DBpedia	93.48%	93.21%	89.23%	90.99%
DBpedia + Translation	93.53%	92.87%	89.24%	90.84%
DBpedia + Translation + Synonyms	92.41%	91.54%	87.46%	89.19%

TABLE IX: DBpedia Approach Results on the DBpedia Exclusive Subset

*3) **Combination of Wikidata & DBpedia**:* Table X shows the model's evaluation scores after the descriptions from Wikidata and DBpedia have been used to extend the data in the Original Data Set before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Wikidata & DBpedia	82.97%	88.55%	79.48%	83.02%
Wikidata & DBpedia + Translation	83.48%	87.86%	79.73%	82.95%
Wikidata & DBpedia + Translation + Synonyms	79.71%	84.63%	75.27%	78.70%

TABLE X: Wikidata & DBpedia Approach Results on the Original Data Set

Table XI shows the model's evaluation scores after the descriptions from Wikidata and DBpedia have been used to extend the data in the Wikidata & DBpedia Exclusive Subset before vectorizing it on a Bag-of-Words representation.

Approach	Accuracy	Recall	Precision	F-Score
Wikidata & DBpedia	92.13%	92.01%	87.98%	89.75%
Wikidata & DBpedia + Translation	92.42%	91.53%	87.64%	89.33%
Wikidata & DBpedia + Translation + Synonyms	91.13%	90.29%	86.24%	87.94%

TABLE XI: Wikidata & DBpedia Approach Results on the Wikidata & DBpedia Exclusive Subset

V. DISCUSSION

To open this section we will briefly introduce what we identify as the most important topics of discussion. We will discuss the results produced with regards to Wikidata and DBpedia as sources of data, the attempt to make corrections for errors introduced, and the best result produced with the linked open data sources. We will explain what their significance is and discuss what implications they might have. By doing this we hope to shed light on what we believe to be the reasons behind why we got the results that we did.

A. *Linked Open Data as Resources*

As we see in Table XII the results produced using the Wikidata and DBpedia approaches show a performance decline compared to the Baseline approach. The observed results indicate that the Baseline approach itself was better suited for training a classification model than the proposed approaches experimented with was. The accuracy of the Baseline approach was **88,60%** and by using the Wikidata, DBpedia, and Wikidata & DBpedia approaches we can observe from Table XIII a decline in accuracy of **3,95%**, **2,12%** and **5,63%**. We also notice a corresponding drop in the other performance measures. As we have shown with the approaches in the previous section and the research presented in II, it is indeed possible to improving accuracy using feature enrichment techniques. Expanding the feature set allows the classifier to find more distinct patterns on which to make decisions. Unfortunately, this was not the case with the data we collected from Wikidata and DBpedia. By further analysis of each linked open data source, we discuss possible justifications for our results.

Baseline	Wikidata	DBpedia	Wikidata & DBpedia
88.60%	84.65%	86.48%	82.97%

TABLE XII: Accuracy for The Baseline, Wikidata, DBpedia and Wikidata & DBpedia Approaches with the Original Data Set

Wikidata	DBpedia	Wikidata & DBpedia
-3.95%	-2,12%	-5,63%

TABLE XIII: Accuracy change of the Linked Open Data Approaches from the Baseline on the Original Data Set

First and foremost, the hit-ratio, denoting how many transactions yielded a match in the linked sources, was relatively small. By counting the number of transactions that produced a result in each linked open data source we noticed that only a little over half of the original data yielded a hit in Wikidata and DBpedia:

- Wikidata Hit-Rate $= \frac{113263}{220618} = 51,34\%$
- DBpedia Hit-Rate $= \frac{125765}{220618} = 57,01\%$
- Wikidata & DBpedia Hit-Rate $= \frac{136474}{220618} = 61,86\%$

Combining the Wikidata and DBpedia approaches was an attempt to increase this hit-rate, but still yielded a relatively low number. The reason for this was the great amount of overlap in

which transactions yielded a match in the data sources. There were as many as **102554** transactions in the original data which yielded a result in both Wikidata and DBpedia. Only **10709** of the transactions were found exclusively in Wikidata and **23211** transactions were found exclusively in DBpedia.

The low hit-rates of all three approaches indicate that the linked open data sources are not extensive enough, separate or combined, for our use and are not likely to contribute positively when training our classification model.

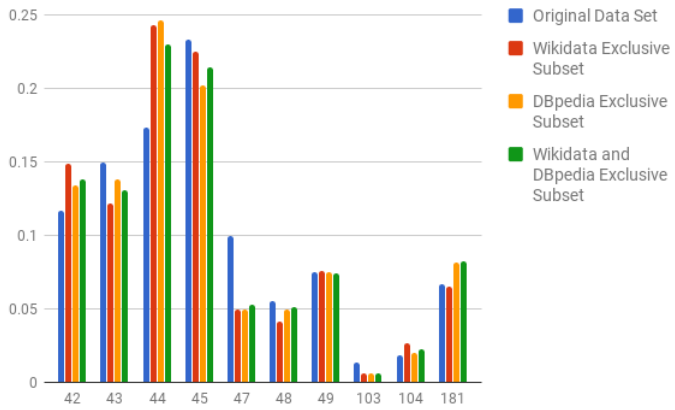

Fig. 10: Normalized Label Distribution over Different Data Sets

Having observed these low hit-rates, we conducted experiments where we used the subsets of the original dataset which contained only transactions which yielded a match in the linked open data sources. We did this to gain insight into how the linked open data approaches could potentially perform compared to the Baseline approach given that all of the original data yielded a match in the linked open data sources.

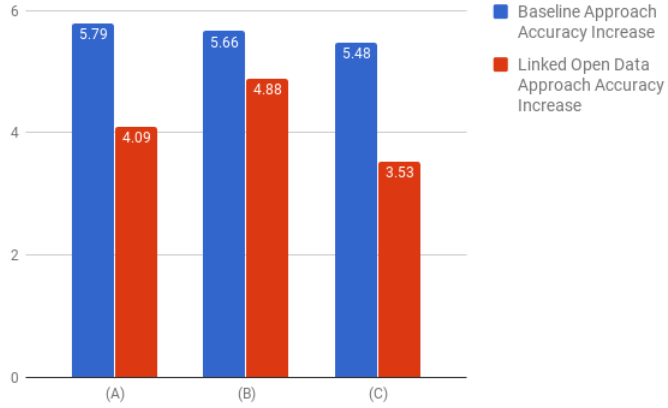

Fig. 9: Accuracy Comparison from the Original Data Set to a Reduced Original Data Set
(A) Comparison of accuracy percentage change of Baseline approach and the Wikidata approach from using the Baseline approach Original Data Set to the Wikidata Exclusive Subset.
(B) Comparison of accuracy percentage change of Baseline approach and the DBpedia approach from using the Original Data Set to the DBpedia Exclusive Subset.
(C) Comparison of accuracy percentage change of Baseline approach and the Wikidata & DBpedia approach from using the Original Data Set to the Wikidata & DBpedia Exclusive Subset.

As we can see in Figure 9, all linked open data approaches increased substantially to the better. However, the performance was still worse in all of the linked open data approaches than in the Baseline approach. This shows that much of the error from using the linked open data sources is introduced by the fact that a lot of the original data does not yield a result in the linked open data sources. We had a theory that the increase in performance could be explained by the subsets having a label distribution which favored classes with a higher recall score. We can, however, see from the label distribution in Fig. 10 that the label distribution, with the exception of the classes **44** and **47**, is approximately the same for the original data set and its subsets. We could, therefore, conclude that the performance increase rather indicates that the removed transactions within each class were harder to classify.

The low hit-rate could be explained by the nature of the bank transaction texts. As stated in III-A, our dataset consists of Norwegian transaction texts where many contain Norwegian company names. This makes it more difficult for us to get results from Wikidata and DBpedia since they contain relatively few Norwegian companies. Both Wikidata and DBpedia are focused on a more general level which makes deeper knowledge on a specific topic hard to obtain from them e.g. companies on a country basis. Smaller companies that operate in only one country are, understandably, not a priority when covering information on a global scale. On the other hand, larger companies and companies that are internationally known tend to give results even though they may be based in only one country. The information that can be extracted from Wikidata and DBpedia seems to be too general for the purpose of this project and does not give information to the extent that we require.

A side-effect of Wikidata and DBpedia covering information on a more general global basis is that the returned information might not represent the correct information. By this we mean that many results are *False Positives* which would make the information we extend the original data with incorrect and misleading. By conducting a Simple Random Sample test for each linked open data source we could see an indication of this. Each Simple Random Sample test consisted of 100 transactions which yielded a result in each of the linked

open data sources. The evaluation was done as a subjective analysis since there is no actual correct answer to this test and therefore results may or may not represent the true results. The sample data of the tests revealed this for each linked open data source:

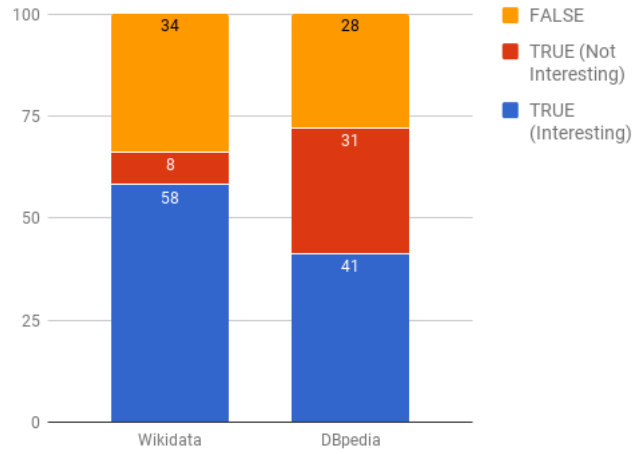

Fig. 11: Comparison of Hit-Value of Wikidata and DBpedia

If we assume that this Simple Random Sample test is representative of the rest of the data that comes from Wikidata and DBpedia, then this is a clear indication that we are introducing many words which do not describe the transactions they are assigned to. We should then expect to observe a performance decline in both the results for Wikidata (see Table VI) and DBpedia (see Table VIII) approaches compared to the results for the Baseline approach (see Table IV). This is also shown in the performance decline in the experiments performed on the Wikidata (see Table VII) and DBpedia (see Table IX) subsets when compared to the Baseline on each respective data set (see Table IV).

From the Simple Random Sample, we see that Wikidata yields a higher percentage of correct and meaningful results than DBpedia since a lot of the results from DBpedia give little meaning. From this perspective, we would believe that Wikidata approach would perform better than the DBpedia approach. However, as we observe in the results, this theory does not hold up, and DBpedia performs a little better. This indicates that extending the transaction descriptions with data that does not contribute to distinguishing between classes, produces better performance results. This suggests that the data from the linked open data sources do not help in this classification problem. These results could also explain why the combined approach performed worse than both the Wikidata and DBpedia approaches because even if one of the linked open data sources return a correct result, the other one may return an incorrect result.

The data returned from both Wikidata and DBpedia was of variable length and content. Two companies that operate in the same industry would often have a description that was written differently, and no standard format was used. This could be another possible source of error. Conformity could have been an advantage for classification since a decision boundary would be more pronounced in the data. The description from Wikidata and DBpedia mainly consists of free-text which makes the description of many companies that operate in the same industry highly variable. This error could also be thought to make the performance of the combination of the two approaches to decline even further, which we believe is another reason for the poor result.

B. *Correction of the Proposed Approaches*

In order to remedy the shortcomings of the linked open data sources, we have used two methods in an attempt to correct this. First, we translate the original data to English and then extend both the transaction descriptions and the data from Wikidata and DBpedia with synonyms.

The translation of the original data showed improvement as seen from the result of all approaches (see Tables VI, VIII and X). We believe that this increase in performance come from the reason that translated original data share more words with the extracted linked open data than with the original data itself. We can observe this effect from the increase in performance when the experiment is conducted on both the original data set and the reduced original data set. The performance increase is true for all proposed approaches. However, even though the there was an increase, the difference was not significant. As seen in the change from Table XIII to Table XIV, the observed improvement obtained by using the translated original data instead of just the original data for the proposed approaches was **0,34%**, **0,17%** and **0,51%** for Wikidata, DBpedia and the combined approaches respectively.

Wikidata	DBpedia	Wikidata & DBpedia
-3.61%	-1,95%	-5,12%

TABLE XIV: Accuracy change of the Linked Open Data Approaches with Translated Original Data from the Baseline on the Original Data Set

Extending the translated original data with synonyms in addition to Wikidata and DBpedia did however not result in a performance increase. As seen in Table XV the performance in the experiments is clearly reduced. We believe that the way synonyms were used to extend the different approaches further contributes to making the problems observed even more significant by looking at the data returned by Wikidata and DBpedia. When we extend with synonyms to create similarities we also, as a side-effect, further reduce the conformity of the transaction texts. This is an effect created by adding many new words to the transaction texts. We also believe that since the data returned might be incorrect, the synonyms only enhance the observed error and therefore also create errors of greater significance. This side-effect was not taken into account when selecting approaches. For these reasons we can see that extending the translated original data and linked open data with synonyms only contribute to a less clear decision boundary to perform classifications on.

Wikidata	DBpedia	Wikidata & DBpedia
-8,73%	-6,86%	-8,89%

TABLE XV: Accuracy change of the Linked Open Data Approaches with Translated Original Data and Synonyms from the Baseline on the Original Data Set

A proposition for a better correction approach would be to replace words rather than just adding them to the text. If word ***A*** and word ***B*** are synonyms, replace them with a word ***C***. See Fig. 12 for an example.

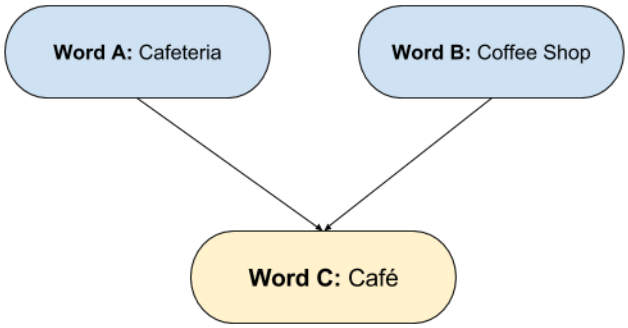

Fig. 12: An Example of Replacing Synonyms with the Same Word

Using the translation approach may also have this effect since it is a possibility that synonyms could be translated to the same word. A more strict filtering method for choosing which words to find synonyms for could also be beneficial since many of the synonyms we extended contributed to confusion when finding a pattern of which we make classifications based on. However, the correction would most likely only result in a small increase in performance since the data of which we extract from Wikidata and DBpedia still is insufficient for use in this project.

VI. Conclusion

Firstly, we can conclude that Wikidata and DBpedia are not fit to be use as data sources in the classification of bank transactions. For a domain like this, consistency and conformity are critical. Due to the nature of linked open data, the granularity of the searches in the linked open data sources is important to get useful information of which we can use to extend the bank transactions with. We found that by using Wikidata and DBpedia, we get very few results on specific domains like businesses, primarily Norwegian, and too much information which either was too lacking or too descriptive to make improvements in our classification problem. A finding in this research, however, is that the data which is possible to extract from Wikidata and DBpedia is better suited for internationally known companies. This means that the results potentially could have been better if the experiments were conducted on a different data set where the companies mainly were internationally known companies and not country specific companies.

Despite our two attempts to correct the shortcomings of the data extracted from both linked open data sources, the yielded results from the attempted approaches were still inferior to the Baseline approach. We believe that with a better approach of how to make use of synonyms we could have produced better results, although, still limited by the quality of the linked open data.

The concept of a structured web is interesting, and using all of this available information shows potential. If the linked open data sources continue to grow and conformity is introduced to the structured data then linked open data may prove useful in projects like this in the future.

References

[1] "Wikidata DBpedia". Availible at http://wikidata.dbpedia.org/. Last accessed 10/06/2017.

[2] Fellbaum, C., "What is WordNet?". Availible at https://wordnet.princeton.edu/ WordNet and wordnets. In: Brown (2005). Last accessed 15/06/2017.

[3] "Natural Language Toolkit". Availible at http://www.nltk.org/, NLTK Project (2017). Last accessed 13/06/2017.

[4] Matt Chaput, "About Whoosh". Availible at http://whoosh.readthedocs.io/en/latest/intro.html#about-whoosh (2012). Last accessed 09/06/2017.

[5] "Yandex". Availible at https://yandex.com/company/general_info/yandex_today/ (2017). Last accessed 30/05/2017.

[6] RDF Working Group, "Resource Description Framework (RDF)". Availible at https://www.w3.org/RDF/ (2004). Last accessed 29/05/2017.

[7] Xiong, C., Callan J. "Query Expansion with Freebase". Proceedings of the 2015 International Conference on The Theory of Information Retrieval, September 27-30, Northampton, Massachusetts, USA (2015).

[8] Christopher M. Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics)", Springer-Verlag New York, Inc., Secaucus, NJ, pp. 205-209 (2006).

[9] Van Asch, Vincent. "Macro-and micro-averaged evaluation measures." Available at https://www.semanticscholar.org/ (2013).

[10] Lovisa B. Skeppe, "Classifying Swedish Bank Transactions with Early and Late Fusion Techniques." Master Thesis. KTH Royal Institute of Technology, Stockholm (2014).

[11] Claudia Perlich, "Which is your favourite Machine Learning Algorithm?" Available at http://www.kdnuggets.com/2016/09/perlich-favorite-machine-learning-algorithm.html (2016).

[12] Erlend Vollset, Eirik Folkestad, "Automatic Classification of Bank Transactions," Chapter 2. Master Thesis. Norwegian University of Science and Technology, Trondheim (2017).

[13] Yoan Gutiérrez, Sonia Vázquez, Andrés Montoyo, "Sentiment classification using semantic features extracted from WordNet-based resources," Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, p.139-145 (2011)

[14] Albitar, S., Espinasse, B., Fournier, S., "Semantic Enrichments in Text Supervised Classification: Application to Medical Domain," Florida Artificial Intelligence Research Society Conference, (2014).

[15] Iftene, A., Baboi, A.M. "Using semantic resources in image retrieval." 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, Vol. 96, Pp. 436-445, Elsevier (2016).

[16] Ye Y., Ma F., Rong H., Huang J.Z. "Improved Email Classification through Enriched Feature Space." In: Li Q., Wang G., Feng L. (eds) Advances in Web-Age Information Management (WAIM), (2004).

[17] Poyraz, M., Ganiz, M. C., Akyokus, S., Gorener, B., and Kilimci, Z. H. "Exploiting Turkish Wikipedia as a semantic resource for text classification." International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 1-5 (2013).