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, Trondheim, Norway
eirik.ek.folkestad@gmail.com, jon.atle.gulla@ntnu.no
[2] Analytics, Sparebank1 SMN, Trondheim, Norway
marius.rise.gallala@smn.no

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. We wish to develop techniques to improve a baseline approach to bank transaction classification 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:

© Springer International Publishing AG 2017
G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 767–780, 2017.
https://doi.org/10.1007/978-3-319-69179-4_54

– 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 the 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 a presentation of the experiments we conducted and the results they yielded. We also discuss our findings in this section. In Sect. 4 we present a few studies which are closely related to the work we are conducting in this project. The paper is summarized in Sect. 5 by summarizing our discussion 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
115603 EURO SKO Dikevn. 28	84	49
Mandal Kommune . Mandal	116	103
TAIGAEN AS . 2340 Løten	74	43
GOOGLE *AbZorba Games	91	48
Til: LM Strømko Betalt: 26.06.13	73	43
XL6000003445	120	181

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. A few examples of entries in the dataset are 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' and $x_2 =$ 'A chair is a chair', the bag-of-words representation produced is shown in Fig. 1a. 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. 1. 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 Fig. 1b.

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 Eqs. 1 and 2).

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.

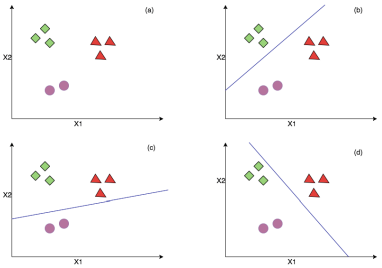

Fig. 2. Logistic Regression OvR example (a) feature-vectors | (b) classifier for diamonds | (c) classifier for circles | (d) classifier for triangles

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 Figs. 2(a), (b) and (c), 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]. In the research we previously conducted on this topic [7] we evaluated Naive Bayes and a Multi-Layer Perceptron. These algorithms were both outperformed by Logistic Regression and are therefore omitted in this paper.

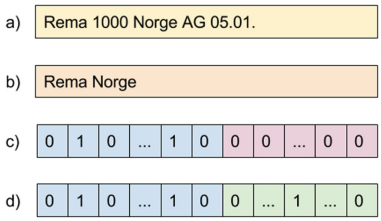

Fig. 3. Transaction representation example (a) Trans. text | (b) Trans.text cleaned | (c) Bag-of-Words w/o Brreg Code | (d) Bag-of-Words with One-Hot Brreg Code

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. 3b). The text is then converted to a vector representation using the Bag-of-Words Model (see Fig. 3c).

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.

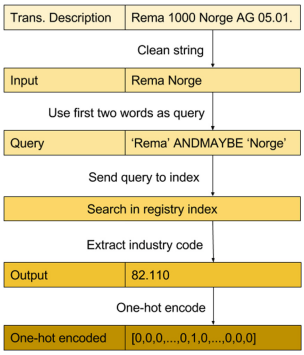

Fig. 4. Industry code extraction example

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 and hence be used as search terms in the indexed entity registry. The transaction description is cleaned in the same way as described in Sect. 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.

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. 3d). 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. 3c). This entire process for extracting industry codes is illustrated in Fig. 4.

2.7 Google Places API

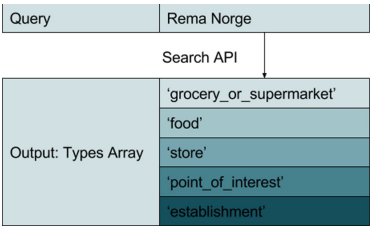

Fig. 5. Google Places API output example

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 Fig. 5. 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 *Brønnyøsund 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.

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.

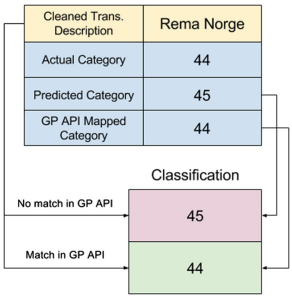

Fig. 6. Google Places API utilization example

This approach is exemplified in Fig. 6 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 and Discussion

3.1 Experiment Description

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 Fig. 7. Here we can see that the accuracy begins to stabilize at size 4,000 making it a reasonable size to use.

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 subcategories, of which there are 63.

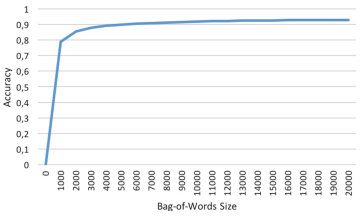

Fig. 7. Accuracy per 1000 increment in Bag-of-Words size

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 Fig. 3c. "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. In the "Combining Approaches" experiment we use both the semantic enrichment techniques.

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 4).

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. 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 %

Table 5. 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

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% 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 bias introduced by the 'dummy' value which is assigned to all transactions without a corresponding industry code.

The label distributions for the exclusive and non-exclusive transaction are approximately the same, which indicates that the baseline results should be the same in both cases. However, if we compare the per class results for the *Brønnøysund Registry* approach in Table 7 and the Baseline in Table 5, 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 (Table 6).

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

Table 6. 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 7. 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

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.

3.4 *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 all of which are shown in Table 8. We refer to these outcomes as classification changes. 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 11, the class contributions, which are weighted normalized differences between negative and positive class changes, are positive for all classes. This means 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 (Table 9).

Table 8. Possible outcomes for Google Places approach

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

Table 9. 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

Ultimately, the *Google Places* approach leads to a 2.01% 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 (Table 10).

Table 10. 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

Table 11. 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

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 (Tables 12 and 13).

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.

Table 12. 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 13. 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

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

4 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.

5 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. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer New York Inc., Secaucus (2006)
2. Van Asch, V.: Macro-and micro-averaged evaluation measures (2013). https://www.semanticscholar.org/. Accessed 23 Apr 2017
3. Skeppe, L.B.: Classifying Swedish bank transactions with early and late fusion techniques. Master thesis. KTH Royal Institute of Technology, Stockholm (2014)
4. Perlich, C.: Which is your favourite Machine Learning Algorithm? (2016). http://www.kdnuggets.com/2016/09/perlich-favorite-machine-learning-algorithm.html. Accessed 10 May 2017
5. The Google Places API Web Service. https://developers.google.com/places/web-service/intro. Accessed 15 June 2017
6. The Google Places API Text Search Requests. https://developers.google.com/places/web-service/search#TextSearchRequests. Accessed 15 June 2017
7. Vollset, E., Folkestad, E.: Automatic classification of bank transactions, Chap. 2. Master thesis. Norwegian University of Science and Technology, Trondheim (2017)
8. Gutiérrez, Y., Vázquez, S., Montoyo, A.: Sentiment classification using semantic features extracted from WordNet-based resources. In: Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, pp. 139–145 (2011)
9. Albitar, S., Espinasse, B., Fournier, S.: Semantic enrichments in text supervised classification: application to medical domain. In: Florida Artificial Intelligence Research Society Conference (2014)
10. Iftene, A., Baboi, A.M.: Using semantic resources in image retrieval. In: 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES 2016, 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.) AIM 2004. LNCS, vol. 3129, pp. 489–498. Springer, Heidelberg (2004)
12. Poyraz, M., Ganiz, M.C., Akyokus, S., Gorener, B., Kilimci, Z.H.: Exploiting Turkish wikipedia as a semantic resource for text classification. In: International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 1–5 (2013)