KU LEUVEN

FACULTY OF SCIENCE

Detecting Flemish Innovative Companies Using Web Scraping

Nusret IPEK

Supervisor: Prof. Jan De Spiegeleer
Department of Mathematics *(Statistics and Risk)*

Cosupervisor: Dr. Michael Reusens
Flemish Statistical Authority

Master thesis submitted in fulfillment
of the requirements for the degree in
Master of Science in Statistics and Data Science

Academic year 2020-2021

Preface

This thesis constitutes my research as part of the Statistics and Data Science masters program at KU Leuven. It presents an application of modern data methodologies to the creation of an official statistic. Hence, the collaboration with Statistics Flanders assisted my goal of presenting a new look into official statistics. This thesis also accounts for the final work in my European Master's of Official Statistics (EMOS) track. The study took place from September 2020 to June 2021 where innovation among Flemish enterprises was investigated.

It was a learning journey that I was both enjoyed and challenged. The study and integration of modern methodologies that deal with big and organic data always attracted me. Instead of being reactive to the developing statistical methods, being proactive will flourish the official statistics domain. In this regard, I would like to thank Statistics Flanders for taking the lead and promoting this innovative thesis.

I would like to sincerely thank both my promoter Prof. Dr. Jan De Spiegeleer and my supervisor Dr. Michael Reusens whose valuable mentorship allows me to complete my work. A special thanks to Machteld Hoskens from KU Leuven and Kris Honraet from VLAIO who helped me with data and interpretation.

Finally, this thesis indicates a change in professional life. The research journey inspired me on my future life goals. I am grateful to my family for their continuous support.

Nusret Ipek

Summary

The development in computational technology enables the application of modern methodologies to big and organic data. The producers of official statistics are experimenting with these new tools in order to construct quality and timely statistics. Nevertheless, the dynamics of new information sources bring new challenges to the process. Due to this, we investigate a particular application of modern techniques where we extract information from business websites.

The traditional survey method to estimate innovation within a given region is Community Innovation Survey (CIS). Compared to the traditional approach, we utilize web scraping, text mining, machine learning and deep learning algorithms. The main focus of the study is to investigate the reproducibility of applications performed in several other EU states. In addition to that, we explore possible state-of-art techniques in order to improve the published results. Lastly, the application concentrates on CIS 2019 whereas we identify various scalability aspects to all Flemish businesses.

The empirical results indicate that the business websites hold valuable information which can be used to classify businesses whether innovative or not. Compared to the baseline study by Statistics Netherlands, we have obtained similar prediction results [10][9]. The traditional pipeline from text processing to binary classifiers resulted in 0.90 accuracy and 0.80 F1 score. On the other hand, the transformers under deep learning techniques achieved 0.91 accuracy and 0.90 F1 score. The significant rise of the F1 score stemmed from the improvement in the recall of non-innovative companies. Despite the increase in evaluation metrics, the deep learning techniques offer considerably less interpretability. In the context of the official statistics, the accountability and transparency of the estimates are essential. Hence, a sub-optimal logistic regression model may arguably be preferred due to its white-box nature.

In the end, this thesis aims to present a complementary methodology in the estimation of official innovation statistics in Flanders. In contrast to the traditional survey approach (CIS), the modern big data application offers timely dissemination of results, cost reduction and no response burden to the businesses. On the other hand, the genesis of the supervised learning models that we evaluated depends on the labeled data from the CIS 2019. Moreover, a census-like application needs further theoretical work which should address model bias and model degradation. Consequently, this thesis intends to lead the way in developing quality and robust statistics for Statistics Flanders by setting the baseline for a scalable approach.

List of Abbreviations

API	Application Programming Interface
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
BTW	Belasting over de Toegevoegde Waarde - id (Value Added Tax - id)
BS4	Beautiful Soup 4
CBOW	Continuous Bag-Of-Words
CE	Cross Entropy
CIS	Community Innovation Survey
CSS	Cascading Style Sheets
DBOW	Distributed Bag-Of-Words
DM	Distributed-memory Model
HTML	HyperText Markup Language
inno5	Aggregated Innovation Label
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
NLP	Natural Language Processing
NSI	National Statistical Institute
RoBERTa	Robustly optimized BERT pretraining approach
TF-IDF	Term Frequency - Inverse Document Frequency
VLAIO	Agentschap Innoveren & Ondernemen (The Agency for Innovation and Entrepreneurship)

Contents

1 Introduction

1.1 Background

1.1.1 Measuring Innovative Companies

Traditional methodologies to obtain official statistics turn over a new leaf. New technologies complement the traditional means of acquiring data and analysis. A prominent example is to automatically classify a company as being innovative or not using their respective website. The conventional way of producing this statistic is by means of a biennial survey, namely the Community Innovation Survey (CIS). Using survey methodology to produce official statistics has several disadvantages. Firstly, it is very costly to conduct a survey nationwide. Secondly, there is an evident response burden for the companies. The response burden can be described as administrative costs. There are even models that attempt to quantify the total response burden. The standard cost model is an example of how the statistical reporting can be an encumbrance not only time-wise but financially as well [53]. After all, the companies are not in business to respond to surveys and they largely perceive surveys as non-value added activities [23]. Besides, it is a long procedure that has a direct impact on the speed of dissemination of the results. Conducting surveys can be considered as a long project that needs to be managed. A survey on this scale would take more than a year to design and tailor [23]. Moreover, the implementation also takes months because of steps ranging from data collection to dissemination. Since new technologies and sources of the data are emerging, we can harness the information cheaper and faster without creating a burden to the companies. Large scale availability of company websites makes it possible to build a predictive model. Still, it is challenging to embrace all companies in a region with a census approach. Nevertheless, the text data on company websites can be used successfully to predict the innovative nature of the companies in several countries [22] [31].

1.1.2 Concept of Innovation

Statistics on innovation are crucial for policymakers to adjust economic decisions given the business conjecture. Furthermore, it is a conceptual variable that all European countries need to monitor periodically. Innovation can be defined as finding new methodologies for both tangible and intangible concepts [4]. For a particular company, it may stem from a new product line while some other company can be innovative by just improving its internal policies. The broad concept of innovation boils down to a binary classification for each company in the CIS survey. Although the survey variable captures different aspects of innovation in a company, the innovation concept is much broader in the literature. In an extensive meta-analysis about indicators of innovation, 82 unique indicators were identified [16]. For this reason, the binary label of innovation from the CIS survey certainly does not cover all points of view on innovation. Due to this fact, it is significant to point out conceptual indicators that are operationalized in the CIS 2019 survey. Traditionally, the binary classification is not just deduced from a single

question in the CIS survey but rather combined from 5 different questions. The underlying concepts of these questions are helpful to understand the fundamental concept of innovation; product innovation, process innovation, abandoned/ongoing innovative activities, and R&D activities. Hence, the meaning of the innovation measured in CIS align with only these particular domains.

Among the other member states, Belgium holds a special position when we investigate the innovative share of the companies. In figure 1.1, the infographic demonstrates that the Belgium has the highest percentage of the innovative companies resulted from the CIS 2016. While the results are extracted from the previous survey, it is the latest infographic available.

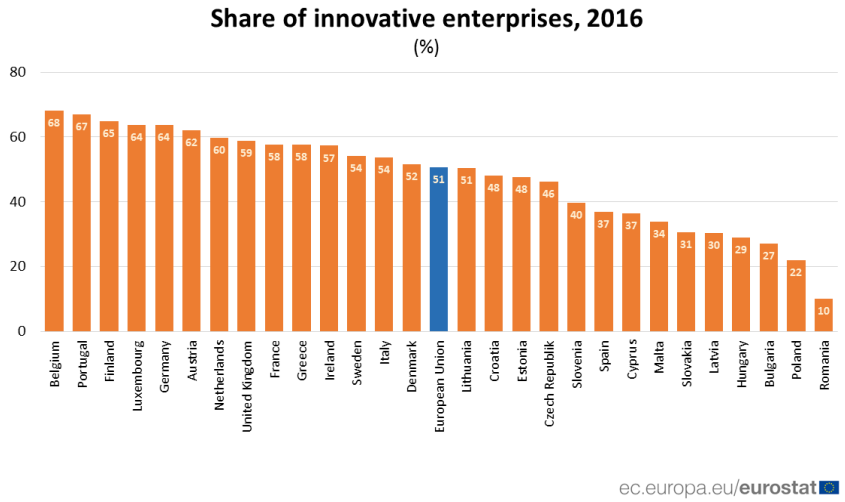

Figure 1.1: EUROSTAT CIS 2016: Share of Innovative Companies in EU [17]

1.2 Web Scraping and Text Mining

Web scraping is defined as a set of techniques dedicated to harvesting web content in an automated way using computer programs [51]. There are many use cases including social network analysis, digital marketing and statistical analysis. In practice, web scraping techniques vary where there is not one single approach that can be applied to all websites. Since each website is built uniquely for different purposes, web scraping techniques need to be adaptive. Therefore, web data mining methodology differs depending on the task rather than a rigorous step-wise procedure to follow. Web scraping is often used in the absence of a custom-built application programming interface (API) that is programmed to provide structured data. For instance, competitor price analysis is an important part of the digital marketing. In most cases, there is no way to access the price data of the competitors using a structured way. Even though there would be few competitors that can provide public API. These are usually not enough to satisfy all the information needs. Hence, web scraping techniques are needed to access and collect data from diverse sources in an organized way. As an advantage, an enormous amount of web data is available for big data analysts to distill statistical information without restrictions of an API. On the other hand, it is a complex and time-consuming approach

depending on the context. The main difficulties in web scraping stem from the fact that most websites are neither designed to provide organized data to a computer program nor willing to fulfill the request from an artificial user. Web pages are also quite dynamic so that the reproducibility of the outputs is a concern. Most of the Flemish companies own a website and these websites will be harvested in order to model the innovative nature of the companies.

In a usual data mining setting, an input of structured numerical data is often used to recognize a pattern and derive information. Until recently, the pattern in the texts left untouched because of the complexity and lack of computational power. Text mining can be defined as a sub-branch of data mining that explicitly focuses on methods and procedures to recognize patterns inside the texts. According to Damerau and others, the methodologies and procedures for mining are not two distinct concepts but share organic similarities [12]. The first step in text mining is to acquisition and extraction text data. Web scraping techniques are ideally suited for the first step in this application but it is not the sole methodology.

Text mining is a developing area of research. Not long ago, the so-called *"Statistical Revolution"* in the area of natural language processing (NLP) emerged [26]. Computational linguistics melt into statistical NLP resulting in state-of-art transformation and analysis of text data. Analogous to web scraping, the application of the steps calibrated according to fit for purpose where there is not a clear-cut approach for all texts. Furthermore, many practical concerns need to be addressed such as storage of the data. Technically, a text is considered as an object in the computer which is harder to access and store compared to numbers. Along with main methods procedures, pointing effective ways to tackle up these shortcomings are crucial. Nevertheless, text mining opens new doors to model patterns in the texts and convert these previously undervalued objects to a valuable information.

1.3 Application to Flemish Companies

As mentioned above, we apply web scraping and text mining methods to produce statistical information about the innovative nature of the Flemish companies. The text data as an input to the models is collected with web scraping techniques. Even so, not all companies possess a web page or at least an accessible one which creates a problem of missing data. Sensitive granular data for each company participated in the CIS 2019 survey is used in the application. There are in total 3,179 Flemish companies that participated in the survey and for each company a binary variable (inno5) indicating its aggregated innovativeness is present. Following the data collection and extraction from the business websites, we model the text from these websites in order to classify the company as innovative or not. Accurate results may generate an alternative to the traditional official statistical production and lead the way for similar official statistics. Henceforth, reproducibility, and scalability of the application are essential. In the end, a successful application to Flemish businesses may decrease the cost of producing innovation statistics, eliminate the response burden, and improve the timeliness of dissemination of the results that is critical for policy-making.

2 Research Questions

2.1 Main Research Questions

In this thesis, the primary objective is to uncover whether an official statistic of innovation can be constructed from the Flemish company websites using web scraping and text mining methodologies. The use of big data and technology to produce official statistics is relatively new in the literature. While some sources label this as an "untraditional" way of producing statistics, these procedures are perceived as merely complementary to the traditional methodologies [46]. There are also unorthodox views that consider the use of big data and technology as an alternative [11]. Although there are limitations of the "untraditional" approach, we believe that new methodology serves as a substitute that measures the same underlying principles of innovation in this context. The research question was explored for other countries previously and revealed accurate results [22] [31]. However, this is the first application for the Flemish region.

2.2 Side Research Questions

Two side research aims are accompanying the main research question.

1. Can the methodology used by other National Statistical Institutes (NSIs) be improved? The added value is to search for state-of-art techniques in the literature for this specific application in order to improve the prediction accuracy.

2. Secondly, this thesis is expected to lead the way for a census approach for all active Flemish companies registered. Consequently, the scalability and the reproducibility of the methodologies are two essential aspects. How can the Flemish Statistical Authority build a census approach upon this thesis? What are the recommendations regarding the large scale implementation of the methodologies? Even though these two questions may sound trivial, many practical challenges need to be answered before moving to a larger scale application such as URL finding, language, and storage of the objects.

3 Related Research

The related research is investigated in two-fold. The first portion consists of similar work done in other regions. The second portion covers the follow-up studies.

3.1 Similar Applications in Other Regions

Although the web mining and structuring its content is not a new field, the idea of using web content to build web-based indicators goes back to no more than 20 years. Katz and others discuss the possible indicators and their robustness but leave the website text content

untouched as an indicator [28]. The work of Youtie and others dives into the website content analysis for nanotechnology companies, however, their application area remains limited to cluster analysis [57]. The study investigates 30 US nanotechnology companies and concludes that there is much potential in the web-based information to model the innovation process. Gök and others present the first complete application using website text to study the abstract notion of innovation [21]. In this paper, researchers extract data and utilize text from individual company websites. Their application groups remain very specific and limited to 296 companies in the UK. Gök and others discuss that the use of website data as an alternative source is significant and useful. Mirończuk and Protasiewicz expand the information domain from only website text to big documents that contain links, descriptions, logo and body text [38]. In this article, Naive Bayes classification is favored when classifying diverse aspects and employ a voting committee of classifiers to label a company innovative or not. In the end, they achieve a relatively high F-measure and find their approach suitable for unstructured website data. Another study that focuses on a sample of 79 Canadian companies reports low but significant correlations between variables of innovation survey and web-based indicators [47]. Kinne and Lenz choose the big data approach in a similar research question [30]. Here, German companies studied whether they are innovative or not solely based on their website texts. Application is limited to a single model under the deep neural networks which reaches a good F-measure (0.8). Furthermore, the study includes granular geographic pattern analysis that is useful in regional policy-making. Lastly, the work of Daas and van der Doef has unique importance because of the similarity in our research aims [10]. Although other studies lead the way of producing a statistical measure from company websites, this study can be considered as the first documented application to create an official statistic under Statistics Netherlands. Consequently, their goal and methodologies are parallel to this thesis. Their models achieve high overall accuracy (88%) and notable F1-score (86%). Furthermore, they performed a census-like application to develop a nation-wide estimate of innovation. As a result of this approach combined with bias correction, the final estimate is very close to the most recent official statistic reported for the large companies in the Netherlands.

3.2 Follow-up Studies

Besides these studies on the classification of innovative companies, there are two critical follow-up studies. The first one is from the Mirończuk and Protasiewicz where their initial methodology evolves to a more complex structure from a classifiers committee to the meta-learning process and genetic algorithm in order to find optimum feature space [40]. In the light of these improvements, they report an increase of 4.6% in F-measure. Secondly, Daas and Jansen published regarding the model degradation of the models derived from the company websites in the Netherlands [9]. Researchers show a dramatic drop in the accuracy of the model by 30% after a year of initial model training. Therefore, retraining the model using updated web content after a certain period is suggested to produce valid estimates.

3.3 Web Scraping

Over the past decades, the web scraping domain adapted many improvements. Many of these improvements stem from the changes in web content delivery standards. For this reason, timely publications are valuable to evaluate the state of art techniques in web scraping. Vanden Broucke and Baesens provide an overview from elementary web scraping techniques to advanced practical challenges [51]. Although the web scraping domain is very vast, this book focus on the applications in data science field. The real-life examples and best approaches to solve them are exceptional contributions for this thesis. In the IEEE conference of Big Data 2019, Diouf and others presented the state of art approaches and tools in web scraping [14]. Here, authors introduce the software platforms and some browser extensions in order to construct best approaches to solve common problems. Since the field is very dynamic, it is important to be up to date with state of art. Dogucu and Rundel focus on the opportunities and challenges of web-scraping specifically in the data science domain [15]. Their work provide additional insights on specific challenges such as relevancy and missing data. Lastly, a knowledge about the inner structure of the web content is crucial. Nearly all web pages are constructed with combination of Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS). The book from Bowers and others give background knowledge in these two web design languages [5]. Although their point of view is constructing a new web page, their methodology is also relevant and useful in reverse engineering them.

3.4 Text Mining

In contrast to web scraping applications, text mining is much more theoretical. The mathematical intuition behind the text analysis plays a significant role to determine the appropriate predictive models. In this regard, Gentzkow and others discuss the text as input data and certain mathematical transformations to make it more suitable for current statistical methods [19]. Feature selection, logistic regression and Richer presentations are worthwhile subjects investigated in this work. The researchers emphasize the growing trend of use of text data in machine learning so that we must be vigilant for state-of-art text mining. The timely article by Mirończuk and Protasiewicz reviews the best practices in text classification [39]. The particular focus on text classification tasks rather than broad text analytics adds an extra value because the predictive models for innovation carry essentially a binary classification task. Here, the researchers divide the methodology into three distinct objectives; extraction, qualitative analysis and quantitative analysis. The state-of-art methodology under these objectives organized with references to a vast literature. For the quantitative analysis, a deeper understanding is necessary both theoretically and practically. "Machine learning for text" book by Aggarwal covers the theoretical background of statistical models in text mining [1]. On the other hand, the practical analysis is done with Python programming language. Therefore, the book authored by Sarkan is favored as a practical guide in text analytics [49]. In this work, the links between theory and practice are established clearly along with our current computational capabilities in NLP.

4 Methodology

4.1 Baseline: Community Innovation Survey 2019

The starting premise of supervised learning models is the availability of label (posteriori) information. The supervised learning models under machine learning depend on previous experience. In our case, this information is defined as a binary variable. This binary variable (inno5) derived from responses to the following domain of questions;

1. Product innovation
2. Business process innovation
3. Abandoned innovation activities
4. On-going innovation activities
5. In-house or external R&D activities

The final CIS dataset contains the inno5 binary variable along with company details such as an address, business registration number (BTW), activity zip code, e-mail and website. The company details filled by the company as part of the CIS and also verified from official data sources such as business register and balance sheets obtained by the Belgian National Bank. We will denote the available information in mathematical terms. The target variable is Y representing the binary inno5 outcome of the CIS. The other available business information will be denoted as M.

$$\{Company\ name\ ...\ Company\ website, inno5\} \rightarrow \{M, Y\} \tag{4.1}$$
$$Y = \{innovative, non\text{-}innovative\}$$

Therefore, the CIS 2019 dataset provided by the ECOOM KU Leuven constitutes the baseline for this thesis. We will use the available information under M in order to classify businesses according to Y. 3,179 Flemish businesses responded to the CIS 2019. There are initially 37 businesses which have null Y. We will disregard these businesses since the supervised learning techniques assume the availability of Y. Among the remaining 3,142 businesses that have a valid Y, there are 2,367 innovative and 775 non-innovative businesses. There is a class in balance where 75.3% of the Flemish businesses are classified as innovative according to the CIS 2019.

Additionally, M contains many missing values. The website information is the most significant feature for this research. Yet, there are 966 business records with missing website information. Hence, we need to manually identify the missing business websites.

4.2 Treatment of the Missing Information

The ultimate information source that will be used is the business websites. So that, we need to have the website information for all the businesses in order to minimize information loss. A valuable source to treat the missing websites is to check the provided e-mail address. The provided contact e-mail usually contains a piece of relevant domain information.

business.manager@companydomain.be $\rightarrow$ http://www.companydomain.be

On the other hand, there is a high correlation between missing website URL information and missing contact e-mail. This method of identification has limits but provides a highly accurate baseline when available. Another possibility is to use search engines to locate the business website with other information contained in M. The most relevant features for precise search engine results are company name and zip code. The search engine methodology has also its own limitations. It is a long and very inefficient process. It would take days to manually search each company with a missing website. Automation of the search process and building helpful indicators for the found websites are considered a good strategy.

We have utilized a custom Python script to scrape Google and Bing search results. The web scraper is built upon the Selenium (version=3.141.0) with the Chrome web-driver. The raw HTML results were parsed with the Beautiful Soup 4 library (version=4.9.1). We have only considered the top 3 returned results. The search phrase is formed by the concatenation of capitalized company name and activity zip code. These two features have no missing values in the CIS 2019 dataset. The found URLs are stored in a CSV file.

Two verification indicators were formulated in order to guide the manual verification. The first indicator is the presence of the zip code in all sub-domains of the returned websites. The indicator is coded as binary outcome whether the website contains a matched zip code or not. The second indicator is the street name lookup in contact and location pages. The Python script visits the given search engine result and searches for the sub-domain links pointing to either contact or location pages. Then, the script extracts all the visible text and search for the street name match. Naturally, this indicator is also binary. These two automated verification indicators are derived from the human verification process. A typical manual verification by a human would be searching the most identifiable information in the website that can link the search engine result to the actual business. A silver bullet to verify a business website is to match its BTW number provided on the website with the CIS 2019 dataset. Unfortunately, not many Flemish businesses provide this information through their websites or they provide it with distinct punctuation patterns. For this reason, it would not produce a reliable automation. The next step for human verification is to use location information such as address and zip code. An exact match of the location information is used to verify the search engine results. By finding the URLs beforehand and building automated checks, a significant reduction of work was achieved.

In the last step of missing data treatment, we manually visited the most promising URLs ranked by the search engine order and two automated indicators until we found a match. The found websites are stored in a CSV file along with a unique identifier (BTW). Despite our efforts, there are numerous businesses that we could not locate a website. Among 966 missing websites, we were able to find 774 websites that we could verify with the information presented. On the other hand, 192 of the businesses that participated in CIS 2019 could not be associated with a website. Since the website text will be used to build predictive models, these businesses cannot be included in our models. Together with the businesses that had null target variable, there are 229 businesses that have crucial missing information. In other words, we have 7.2% missing information that we cannot treat. These businesses will be disregarded when we build predictive machine learning models.

There are certain limitations to this approach of missing data treatment. First of all, the procedure is still time-consuming. The workload increases linearly as the missing business websites increase. Considering the fact that there are over one hundred thousand businesses in Flanders, a complete search would be infeasible. Hence, the scalability of this procedure needs a better strategy to harvest missing data. Secondly, the final decision of associating websites to the CIS data is a subjective decision. There are borderline cases where the website can equally be associated with the business or not. In that case, the decision can lead to both false positive and false negative errors. A rigorous way to avoid a subjective decision is to use other information channels such as direct contact.

The governmental bodies currently working on the best methodology to build an extensive dataset on business domain addresses. After successful completion of the project, all the Flemish businesses will be linked to a URL or missing indicator. In other words, a census-like approach without having trouble to treat the missing data can be possible in a near future.

In conclusion, the partial missingness of the CIS dataset was treated with the aid of novel automated indicators. While the combination of automated indicators and manual checks was feasible for the CIS 2019 dataset, the scalability to all Flemish businesses needs refinement. The websites will be used as the sole predictors and their validity has utmost importance.

4.3 Web Scraping and Storage

4.3.1 Web Scraping

In the next step, we utilized the Python Requests library (version=2.24.0) to collect the raw HTML of the main domain. Application in the Netherlands by Daas and van der Doef showed that the sub-domains are not useful to build predictive models [10]. Consequently, only the default landing pages of the businesses scraped. The SSL certificate check was disabled since numerous businesses owned websites with an expired SSL certificate.

There were several websites that could not be accessible during our initial attempt. Due

to maintenance and network problems, it was an expected behavior. In order to maximize the information gain, we tried to access those websites two more times with a one-week interval in between attempts. Following the three web scraping attempts, 30 business websites could not be accessed. Most of the inaccessible websites were not available anymore and return HTTP request codes of 400, 403 and 503.

One of the unique features of the Flemish websites is their multi-lingual property. Belgium has three official languages; Dutch, French and German. Additionally, English is one of the mainly used languages by businesses. The homepage language is an important factor for text mining since every language has its characteristics. Hence, a choice of a common language would lead to better predictive models. On the other hand, the diversity of the websites does not allow to concentrate on a single language in the Flanders area. Many business websites offer either Dutch or English along with other languages. Consequently, we will build our models using both Dutch and English languages and prioritize the Dutch language if available.

4.3.2 Language Switching

Since the website determines in which language it will display the homepage, we need a procedure to adjust the choice to our predetermined language selection. The usual human interaction with a website would be a search for the language menu. This language menu can be implemented with various visual cues such as flags, language abbreviations or plain language names. In any case, the raw HTML file would contain the coded information for that switch menu if available. Similar to the manual missing data treatment, it is a laborious procedure to switch each non-Dutch website manually.

In the literature, there is no application of automated language switches. A naive approach presented in the web scraping books uses GET request functionality to request a website in a particular language. While sending the GET request to the server, we can set the 'accept-language' parameter to 'nl-be'. This would ask the server to return the Dutch content with Belgium locale settings. Although it is a simple attempt to switch the language, the web servers usually do not fulfill the request. Many business websites have been developed with low budgets and are not configured for unusual requests. Moreover, there is no uniformity in the website designs and web servers. The highly unstructured format of the websites creates difficulty to program robust manipulative scripts such as a language switcher.

Our novel algorithm starts by searching the 'hreflang' attribute which is optionally used in predefined <a> tags in the HTML files. This attribute codes the language of the destination URL contained within the tag. The web developers use <a> tags for search engine optimization since it allows semantic processing of localized results. Due to the common utilization of this attribute, we can use the localization codes to extract the most probable URLs pointing to the Dutch version of the website. If a website HTML code does not use the 'hreflang' attribute then we need to investigate all links contained on that page. The found links within the HTML

code checked for the language content using the TextBlob library (version=0.15.3). Since a website can contain links to an external website as well, we built safeguards to remain in the input domain. The simplified pseudocode of the language switching is given in Algorithm 1.

Algorithm 1: Python Language Switcher

Input: URL, Language
Output: Success Code, New URL
begin
  for $hreflang \in URL$ **do**
    if $DetectLanguage(hreflang) = Language$ **then**
      return True, $hreflang \rightarrow href$
    end
  end
  for $href \in URL$ **do**
    if $Language \in lower(href)$ **then**
      if $DetectLanguage(href) = Language$ **then**
        return True, $href$
      end
    end
  end
  return False, ' '
end

4.3.3 Data Storage: ArangoDB

Structured and scalable storage of the scraped websites in logical structure together with the baseline M is important for maintainability. ArangoDB is an open-source database system that will be used as primary storage of the dataset and HTML files. The ArangoDB is a multi-model database that has support for the documents. The community edition (version=3.7.2) of the ArangoDB is used to create a document-based database. Then, both M and Y are transferred to the unique documents. The BTW number of each business is used as the primary key. The main advantage of ArangoDB is the availability of a Python driver. The documents can be accessed directly with a Python script and manipulated using custom functions.

HTML files are essentially textual files. They combine web scripts, pre-determined tags and visible content of the website. The HTML files are human-readable but optimized for machine interpretation. The textual aspect of these files allowed the storage of the contents as part of the database document objects without requiring an explicit file management system. This relational database structure provided logical structure and order to the enhanced dataset. Furthermore, the database management system secured sensitive official information (granular linked information).

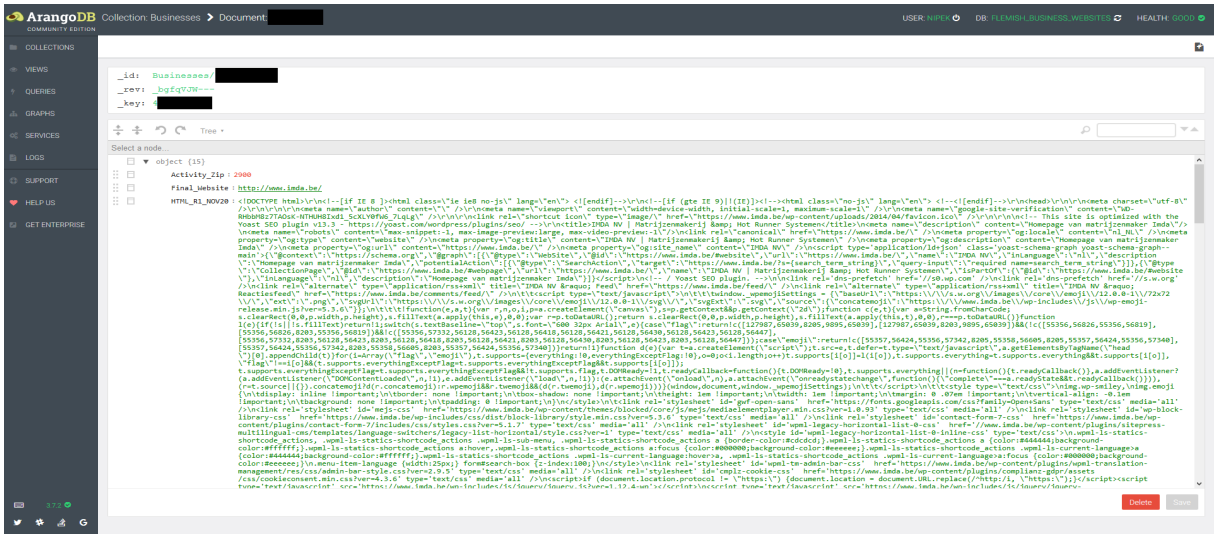

Figure 4.1: ArangoDB: Interface and Random Document Representation

In conclusion, the URLs provided by the surveyed companies combined with the URLs found by the missing data treatment strategy and passed to the web scraping script. Instead of a loose collection of CSV and HTML files, we opted for a tight collection of document model that has additional secure access benefits. Both the web scraping approach and the storage aspect are scalable for census-like analysis of all Flemish businesses.

4.4 Text Processing

4.4.1 Extracting Visible Text

The HTML files are designed to display the website content and interact with the users. A specific HTML code may include predefined tags, attributes, header, metadata, scripts and many other markup language syntaxes. The visible text on a website is only a small fraction of the raw HTML file. The computer syntax does not contain valuable information to classify a company as innovative or not. For this reason, we need to extract only the visible text from the harvested HTML files. Given that, the design of the website and even the media content can be considered during the classification, whereas these advanced features left out-of-scope due to the limited applications in the literature.

The default HTML parser under the BS4 library was used to create a parsed HTML documents. The tags are semantically labeled in a tree structure. Then, the labels associated with the script, head, title and style tags have been removed from the tree. Secondly, the cookie and GDPR pop-up messages filtered out using special keywords. Lastly, some of the HTML files included <iframe> tags. The <iframe> embed HTML code stored in other domains into the scraped page. Consequently, the visible text is actually requested from other URLs on the run-time which we did not scrape. For those tags, we scraped the <iframe> source contents as separate HTML files. The visible text extracted from those frames is appended at the end of the visible text extracted from the original URL.

The identifying information (M) converted to a pure text representation of the businesses. The visible website texts establish the backbone of the business representation. The notation of this representation is denoted as R. We will use the information contained within the R to build predictive models.

$$\{M, Y\} \rightarrow \{R, Y\} \tag{4.2}$$

4.4.2 Language Detection

Every multilingual text processing should start with the identification of the document language. It is a piece of crucial information for the proceeding text mining techniques. Many text-processing steps are language-dependent such as stop word lists, stemming and lemmatization. We have already used the language detection mechanisms during the language switching algorithm. Put it differently, we caused the exchange of homepage URLs to Dutch and English by using TextBlob library. Nevertheless, numerous websites could not be switched to Dutch or English.

The web scraping phase iterated over the best possible URLs without considering the source language. Consequently, the database contains documents with undesired HTML instances. We need to detect the language of these HTML files properly before the natural language processing phase. The TextBlob library employed to detect the language on the visible text of the final scraped HTML files. The TextBlob library uses the Google Translate API to determine the source language. Consequently, it is more accurate compared to the deterministic language detection algorithms which use trigrams.

Some of the extracted visible texts from the HTML files are too short for proper language detection. For those pages, we labeled the language as unknown. The business websites classified other than Dutch or English disregarded since the methodology depends on the language semantics.

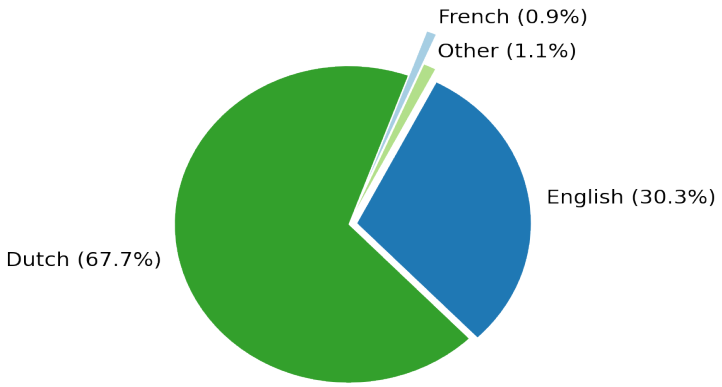

Figure 4.2: Distribution of Final Website Languages

Figure 4.2 shows the distribution of the website languages. As expected, the share of the Dutch language is the highest and it is followed by English. Despite our attempts to switch, there is a small portion of the websites categorized as French and others (2%). The omission of these businesses increases the total missing values to 9.97%.

4.4.3 Natural Language Processing

The high-level natural language processing pipeline consists of tokenization and normalization of the raw text [3]. On the low-level processing, there are impactful decisions to make that have a direct effect on the final model accuracy. The textual data analysis focus on the raw texts from various sources and each source has its own characteristics. The natural language processing of the website texts is still open for optimization with low-level NLP. We will follow the low-level processing since it will allow us to present the effects of different decisions on the final model accuracy for the website data. Because of a bilingual aspect of the R, we built two separate pipelines with identical steps.

The initial processing phase removes the punctuation and converts the lowercase characters. The removal of the punctuation is done with regular expressions. The regular expressions are deemed to be extremely powerful textual pattern finding tools and widely used in the low-level processing [18]. The following regular expression is used to extract only the alphabetical characters globally and eliminate all the punctuating and numerical characters.

$$/[A - Za - z] + /g \tag{4.3}$$

Secondly, the special letter sequences used in websites indicate the necessity of low-level text processing. Among the website visible texts, it is common to see numerous top-level domain extensions like 'be', 'de', 'com' and etc. These short abbreviations do not contain contextual information regarding the innovative nature of a business. The application on the businesses in the Netherlands showed that the short sequences are over-weighted and jeopardize the models [10]. Hence, we eliminated the character sequences with three characters or less. The disadvantage of this elimination is the contextual information loss from the short words.

Thirdly, every language has specific words used as conjunctions, prepositions and other widely used key phrases. The omission of the stop words practiced widely in the text mining domain [29]. On the other hand, some studies conclude that removal of the stop words has minimal effect on the final accuracy [20]. The effects of stop words can be judged with the sensitivity analysis. During the initial model fitting, we removed the stop words by using the pre-built lists within the NLTK library (version=3.5).

Next, the stemming and lemmatization of the remaining words are considered. In order to increase the accuracy of the models, many applications of text mining consider stemming.

The stemming of the words refers to the extraction of the root form of a word so that during the transformation phase these words are represented identically. An example of the stemming in English;

$$\begin{aligned} text &\rightarrow text \\ texts &\rightarrow text \\ texting &\rightarrow text \\ textual &\rightarrow text \\ textually &\rightarrow text \end{aligned}$$

Similar to the stemming, a lemmatization process looks for the common part of the different words (lemma). The difference is that lemmatization groups by contextual meaning. For this reason, it is more advanced form grouping compared to the stemming algorithms. The difference between the stemming and the lemmatization algorithm can be presented with a simple example;

$$Stemming \rightarrow \begin{cases} bad & \rightarrow bad \\ worse & \rightarrow wors \\ worst & \rightarrow wors \end{cases} \qquad Lemmatization \rightarrow \begin{cases} bad & \rightarrow bad \\ worse & \rightarrow bad \\ worst & \rightarrow bad \end{cases}$$

During the model search, we do not use the stemmed or lemma form of the words. Most websites hold limited number of information that is presented with a short text. The stemming and lemmatization becomes more meaningful for the longer texts. Nevertheless, the effect of the stemming and lemmatization is evaluated during the text processing sensitivity analysis.

Lastly, some of the websites contain very limited information which is not enough for accurate classification. For instance, a website scraped during the maintenance phase will only show information regarding the maintenance phase. Similarly, a web page with an expired domain would display only the notice of expiration. The detection of these practical problems is not trivial. Nonetheless, we have to consider only the websites that encapsulate relevant information for the classification task. The application to the website in the Netherlands reports 15 words as the minimum for a quality classification [10]. We increased the suggested minimum word threshold to 25 words for better quality since most expired domains contained more than 15 words but less than 25. The remaining businesses after the text processing reduced to 2,746 compared to the initial 3,179 surveyed companies. After the final removal of the businesses' websites with short visible text, the total coverage is 86.38%. Considering our

position on the importance of input data quality, the coverage appears to be remarkable. The application in Netherlands case reached a coverage of 75.13% after the text processing [10].

4.5 Oversampling & Transformations

4.5.1 Oversampling

A prominent problem in many machine learning applications is a class imbalance. The figure 1.1 suggested that Belgium has an exceptional position within other European countries. The innovative share of the businesses is largest in Belgium according to the CIS 2016. Furthermore, the initial exploratory analysis showed that the reported share increased in CIS 2019. Because many classification algorithms perform well under the balanced classes, we have to look for strategies to minimize the problem.

Re-sampling, down-sizing and learning by recognition are considered as viable strategies to solve the class imbalance [25]. Moreover, the ensemble learning techniques harness the power of the random re-sampling strategy. As a side effect, the class imbalance problem is managed without additional steps. In the application to Flemish businesses, we used the oversampling strategy. The representation of the words within the non-innovative websites has a lower probability when we transform the textual data to the mathematical format. For this reason, the oversampling of the non-innovative websites before building the transformation solves the class imbalance problem. After the oversampling strategy, the innovative share of businesses in the dataset balanced from 78% to 51%. Put it differently, we sampled the non-innovative websites randomly with replacement until a balance has been achieved.

On the other hand, the oversampled businesses should only be considered when we transform and train the data. The test dataset should never contain a duplicate business. Hence, the businesses with duplicate unique keys were removed from the dataset after the transformation phase. Because the oversampling strategy worked well, other solutions suggested were not considered.

4.5.2 Document Vector Representations

4.5.2.1 Bag of Words (BoW)

The R space contains the processed texts with the aforementioned steps. In order to fit the traditional predictive machine learning models such as logistic regression and random forests, we need to represent the textual documents with mathematical vectors. The traditional supervised learning algorithms cannot handle the raw text input as predictors and need transformation. One of the simple vector representations is the bag of words (BoW) [58].

The algorithm is intuitive but has certain limitations. Let each website has a processed text contained in $R = \{r_1, r_2, \ldots, r_i\}$ where $i = [0, 2746]$. The bag of words algorithm transforms the R to $X = \{x_1, x_2, \ldots, x_i\}$ where x_i represents the document vector representation of the

i^{th} website. Since x_i is a vector, the represented words inside that vector can be denoted as $\{w_1, w_2, \ldots, w_j\}$ where the j represents the size of the vector. Each w variable is the normalised occurrences that particular word in a given x.

In practice, there are two most commonly used normalized occurrence representations; binary and count vectorization. In the binary case, the appearance of a w_j in a x_i turns the binary switch. Hence the possible parameter space for $w_j = \{0, 1\}$. The word is either inside the vocabulary of r_i or not. Secondly, the count vectorization includes more information compared to the binary vectorization. The possible parameter space for w_j under the count vectorization is $\mathbb{Z}^+$. In contrast to the binary representation, w_j is counted throughout r_i and the occurrence count reflected to the x_i vector [59].

A large text corpus implies a large vocabulary. For this reason, the representation of all words inside the vectors of X would lead to a very sparse matrix. The j which indicates the size of the document vector representation would be equal to the unique vocabulary count. Since we have a limited number of websites to train upon, we had to avoid the curse of dimensionality. The processing of the text helps to decrease the word space. For example, the removal of stop words shortens the vocabulary and lowering the case avoided the multiple representations. We can also denote the advantage of stemming and lemmatization where the root (or lemma) form of words diminish the vocabulary size significantly. Yet, the reduction does not imply better predictive models. A better strategy to avoid the curse of dimensionality is to choose j as a hyper-parameter and keep the words based on the total observed frequency. A simple bag of words example with $j = 4$;

$$R = \begin{cases} r_1 = \text{"Alice likes math and science classes"} \\ r_2 = \text{"Algebra and science are most important classes"} \\ r_3 = \text{"Alice and Bob skipped the classes today"} \end{cases} \tag{4.4}$$

The vocabulary consists from {classes, and, science, Alice, today, the, skipped, most, math, likes, important, Bob, are, algebra} which is sorted based on the frequency of the words in the R vocabulary. It is apparent that even a short document can create a large vocabulary.

$$X = \begin{cases} x_1 = \{1, 1, 1, 1\} \\ x_2 = \{1, 1, 1, 0\} \\ x_3 = \{1, 1, 0, 1\} \end{cases} \tag{4.5}$$

In this particular case, each vector has a unique representation of the initial texts. Since there are no repeating words inside the text specimens, the binary and the count representa-

tion lead to the same document vector representation. There are several limitations of using BoW features. First of all, due to the large size of the corpus vocabulary, we need to prune the vectors. The supervised learning models cannot handle the cases where the number of features (j) is larger than the labeled document size (i). The omission of the less frequent words from the vector representations accompanies the information loss and still be a sparse matrix. The sparse matrix creates problems for pattern finding in a large predictor space. Furthermore, the semantics of the words are not considered in this transformation [1]. In the above example, the math and algebra words have semantically close meanings. Yet, the BoW fails to create associations between these two words and treats them independently. This is the direct result of ignoring the word order and the context. Lastly, the document lengths are highly variable in practice. A corpus may contain documents including some with few words and others with thousands of words. The vocabulary formation and selection based on the word frequencies become biased toward the large documents. Given the limitations, the BoW vector representation is very intuitive and easy to apply. Consequently, it is widely used in the practice as a method of feature extraction from text documents. We have used the implementation under the scikit-learn library (version=0.23.2) [43].

4.5.2.2 Term Frequency - Inverse Document Frequency (TF-IDF)

Due to the limitations of the BoW transformation, we also consider other document vector representations such as term frequency-inverse document frequency (TF-IDF). The principal idea of the TF-IDF is to use a weighting scheme that can enhance the representation of words according to the local and global word frequencies [56]. Although the weighting scheme of the TF-IDF has published long ago, it is still the most preferred approach on some domains [2] [27].

In order to examine the document vector representation, we can deconstruct the term into two parts; term frequency and inverse document frequency. Let an single textual document $r \epsilon R$ and a word $w \epsilon X$. The raw count of the words suggests the BoW transformation with counting, $c(w, r)$. The term frequency adds an additional weighting factor to the raw count. The raw counts are divided by the total number of the words in r.

$$\text{Term Frequency} \rightarrow tf(w, r) = \frac{c(w, r)}{\sum_{w' \epsilon r} c(w', r)} \tag{4.6}$$

The term frequency specifies the local relevance of a particular word for r. In other words, we scaled the presence of w which is bounded between 0 and 1. Using the term frequencies, the document vector representation accounts for the longevity of r. Hence, the resulting term frequency ranking is robust to the words contained in the lengthy documents. In the second portion of the TF-IDF, the inverse document frequency is an indicator of the rarity

of w. Let $c(w, R)$ be the total count of the documents in R where w occurs and $|R|$ be the total number of documents in R.

$$\text{Inverse Document Frequency} \rightarrow idf(w, R) = \log\left\{\frac{|R|}{c(w, R)}\right\} \tag{4.7}$$

The TF-IDF of w in document r is the multiplication of the two components. This summarizes the overall approach but there are variations on both term frequency and inverse document frequency calculations. For instance, the scaling of the term frequency operation can be replaced with a log-normalization in some applications; $(\log c(w, r) + 1)$. Although there are different variations, we used the standard implementation in the scikit-learn library presented in formula 4.8. This version includes a smoothing tune for the inverse document frequency and euclidean normalization of vectors [43].

$$\begin{aligned}
&\text{Inverse Document Frequency} \rightarrow idf(w, R) = \log\left\{\frac{1 + |R|}{1 + c(w, R)}\right\} + 1 \\
&\text{TF-IDF}(w, d) = tf(w, r) * idf(w, R) \\
&\text{v}_{l2-norm} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}}
\end{aligned} \tag{4.8}$$

Similar to the BoW strategy, the TF-IDF is vulnerable to sparse matrix output. In our application, we limit the number of words chosen to 400 words. The effect of the number of words analyzed in the Netherlands application and the optimal result achieved with 400 words [10]. Even though the conceptual selection of the most relevant top-n words similar to the BoW transformation, the TF-IDF can be used to query documents with maximization illustrated in formula 4.9 [45].

$$\text{argmax}_R \sum_i w_{i,r} \tag{4.9}$$

A closer look into the mechanics of the formulas reveals some interesting points. First of all, let w_i be a very common word among all documents of the corpus. Then, the occurrence of the denoted as $c(w, R)$ will be close to the number of documents ($|R|$). This would bring the inverse document frequency close to 0. Put it differently, the common words in the corpus down-weighted since there is no contextual information and discriminatory power. During the natural language processing phase, we eliminated stop words by using NLTK lists. The TF-IDF is capable to adjust the weighting of these words by default so that there is a minimum value of pre-removal when we use TF-IDF. Conversely, let w_i be a rare word. Then the denominator of the inverse document frequency function will be small compared to the numerator. This would imply $log(x)$ where x is a large positive integer. It means that w_i has a contextual value in the corpus. In sum, the weighting scheme of TF-IDF is elegant and easy to interpret. Compared

to the document vector representation of BoW, the TF-IDF is capable to handle contextual relevancy.

While the TF-IDF improves the predictive power of the models, it has certain limitations as well. The example corpus created in formula 4.4 included semantically close math and algebra words. The BoW document vector representation disregarded the semantic connection between those words and treated them independently. The TF-IDF transformation also cannot account for the semantic relationships. Consequently, the resulting vectors do not account for the relationship between similar words [45].

In vectors presented in formula 4.10, the most frequent words transformed from example 4.4. These words were $\{classes, and, science, Alice\}$ respectively. Since all words in the corpus are unique for each document, it is easy to track the mechanics. The $\{classes, and\}$ words globally occurring in more documents compared to the $\{science, Alice\}$. Hence, the first document vector transformation for r_1 assigns more weight to the $\{science, Alice\}$ words as expected. In r_2, the length of the sentence is taken into account by term frequency portion and the weights are adjusted accordingly. Overall, the TF-IDF has advantages over the simpler BoW approach and easy to compute. Nonetheless, the semantics of words are still not valued.

$$X = \begin{cases} x_1 = \{0.4337, 0.4337, 0.5585, 0.5585\} \\ x_2 = \{0.5228, 0.5228, 0.6733, 0.0000\} \\ x_3 = \{0.5228, 0.5228, 0.0000, 0.6733\} \end{cases} \tag{4.10}$$

4.5.3 Embeddings

In NLP, the embeddings refer to the representation of contextual meanings. The document vector representations of BoW and TF-IDF cannot account for the semantics so that we need embeddings to extract semantic knowledge from the website texts. In the Netherlands application, the Word2Vec approach is included along with TF-IDF to improve the model accuracy [10]. We consider both Word2Vec and Doc2Vec embeddings which are extensively used in NLP.

4.5.3.1 Word2Vec

The Word2Vec algorithm has been developed by Tomas Mikolov in 2013 [37]. The essential idea of the word embeddings is to illustrate each word in the corpus with a vector given a pre-determined fixed size. The vector of similar words such as algebra and math are expected to be mathematically close to each other. There are two common algorithms to train a Word2Vec model. The first one is a skip-gram model which tries to predict the neighboring words (context) given a word. Oppositely, the continuous bag of words (CBOW) algorithm tries to predict the word given the window of neighboring words.

Both algorithms are trained on a shallow neural network setup. In our application to Flemish websites, we used the CBOW algorithm combined with a window size equal to 5 (default value). The Gensim implementation in Python is used (version=3.8.3). The figure created by Mikolov and others encapsulates the two algorithms [36].

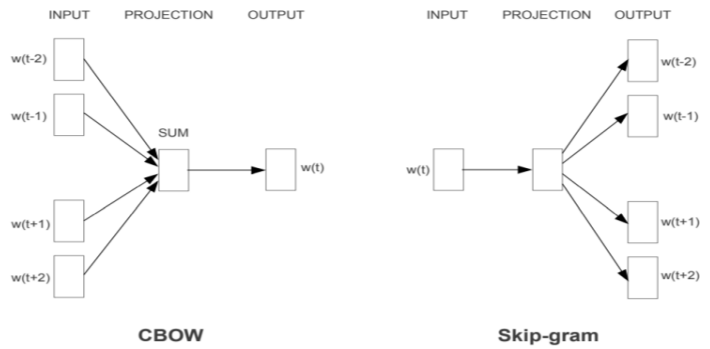

Figure 4.3: CBOW & Skip-gram Algorithms [36]

Both algorithms can either use negative sampling or hierarchical softmax schemes. In our implementation, we opted for the hierarchical softmax since it has advantages on training efficiency [48]. Therefore, we can formalize the probability function of observing a word given a context window as in formula 4.11 [36]. Moreover, one can derive the loss function of the CBOW model under hierarchical softmax which is illustrated in formula 4.12 where S is defined as vector size [42].

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\Big(\big[n(w,j+1) = ch(n(w,j))\big].{v'_{n(w,j)}}^T v_{w_I}\Big) \tag{4.11}$$

$$\begin{aligned} L_{CBOW} &= \sum_{i=1}^{|R|} \log p(w_i|c_{i-j}\dots c_{i+j}) \\ &= \sum_{i=1}^{|R|} \log \frac{e^{\frac{1}{2j-1}\sum_k c_k^T w_i}}{\sum_{s=1}^{S} e^{\frac{1}{2j-1}\sum_k c_k^T w_s}} \end{aligned} \tag{4.12}$$

The relationship between word vectors can be expressed in mathematical terms as well. Assuming the similar words hold a place in the vector space with fixed dimension, two closely related word vectors should be mathematically close to each other. The prominent metric to measure how semantically close the resulting word vectors is the cosine similarity. The cosine similarity function is given in formula 4.13. The interpretation of the cosine similarity is straightforward where $cos(w_1, w_2)$ is bounded between 0 and 1. Cosine similarity is equal to 0 when the word vectors are orthogonal to each other. In other words, they are semantically completely

unrelated. On the other hand, the semantic proximity increase as the $cos(w_1, w_2)$ goes to 1. The math and the algebra words would be expected to have high cosine similarity.

$$Cosine\ Similarity = Cos(w_1, w_2) = \frac{w_1.w_2}{||w_1||x||w_2||} \tag{4.13}$$

Although the word vectors are useful to project word semantics to a vector space, their integration with the document vector representations is not immediately apparent. The word vectors cannot be added directly to the document vector representations since we have no longer at the document level of features. The output of the TF-IDF is structured such that each row is a website text document and each column is a feature derived from the most frequent corpus words. Unlike TF-IDF, the word vectors are unique for each word in the corpus. In order to integrate the word embeddings, we used averaging strategy. The words in a website document are exchanged with the corresponding word embeddings and then the average of the vectors is calculated.

The usage of word embeddings allows extracting semantic information from the words. In practice, the word embeddings increase the predictive power of the models by encapsulating the information that cannot be used by the traditional document vector representations. On the other hand, the deep learning aspect and the integration to the document-wise features place the algorithm close to the black-box models. The high-level understanding of the algorithm does not always provide sufficient interpretation of high-dimensional vector parameters easily. Additionally, the word embeddings need a training corpus and its effectiveness is correlated with both the quality and size of this corpus. We trained the word embeddings using the scraped websites corpus which may not be large enough to extract certain semantic knowledge.

4.5.3.2 Doc2Vec

The word embeddings constructed with the Word2Vec algorithm is extended by the Doc2Vec [34]. Since the word embeddings had document-wise integration problems, the Doc2Vec uses the Word2Vec algorithm to uniformly create document-wise vectors. These vectors can also be applied to paragraphs. In Doc2Vec, both document vectors (D) and word vectors (W) are trained simultaneously. Consequently, the contextual word mapping barrow information from both vectors [34]. The strategy to upscale the word embeddings is similar to our manual implementation which uses averaging and concatenation.

Similar to the Word2Vec models, the Doc2Vec model has two prominent algorithms. The skip-gram in Word2Vec translated as a distributed bag of words (dbow) and likewise, CBOW translated as distributed memory model (dm). We used the Gensim library (version=3.8.3) for the Doc2Vec implementation. The distributed memory algorithm is the default choice because

of its performance. The other hyper-parameters inherited from the Word2Vec model such as minimum word count (2) and window size (5). Given their hierarchical relationship and default values, the choice of inheritance can be justified.

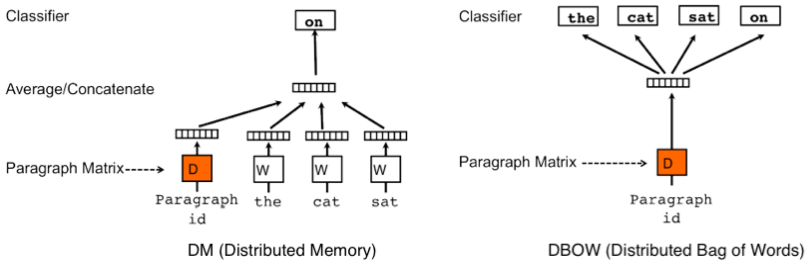

Figure 4.4: DM & DBOW Algorithms [34]

The figure 4.4 taken from the paper of Lu and others elucidate both algorithms with the averaging/concatenation strategy. The main advantage of Word2Vec and Doc2Vec is being an unsupervised learning technique. The training does not require a labeled data input so that the improvements are possible with additional unlabelled business websites. Furthermore, the document embeddings increase the model performance by overcoming the key weakness of the traditional document vector representations which is the lack of contextual learning. The Doc2Vec provides a uniform setting to produce document-wise contextual features in collaboration with word embeddings. The concise document-level representation of semantics can easily be integrated with the traditional document vector representations. The empirical studies indicate that the Doc2Vec improves prediction accuracy [33].

4.6 Classification Models

The transformations of the raw website texts into mathematical feature vectors allow us to fit traditional supervised learning techniques. The R transformed into X where it contains both document vector transformations and contextual embeddings. The resulting structure is a tabular data; each row is a business website and each column is either a frequent word or embedding score. Furthermore, we have the label information which was denoted as Y. Among the traditional binary classification algorithms, we experimented with logistic regression and random forest.

4.6.1 Logistic Regression

The logistic regression is a white-box model which permits direct interpretation of the model parameters and properties. The logistic regression holds a special place in the generalized linear models because of its relationship with Bernoulli distribution. Naturally, the innovative nature of the Flemish businesses is labeled with a binary outcome. We can write the logistic regression model with the following formula.

$$E(Y|X) = \pi(X) = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_p x_p}} \tag{4.14}$$

This default non-linear transformation is known as the logit transformation. This transformation has desirable properties such that the expected value does not range from $[-\infty, \infty]$ but remain in the $[0, 1]$ [24]. It turns out that the $E(Y|X)$ can be interpreted as the probability of the Bernoulli trial. The logistic function can be converted to linear predictors with the following transformation.

$$f(x) = \ln\left(\frac{E(Y|X)}{1-E(Y|X)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{4.15}$$

Secondly, the parameter estimates are interpreted as log odds ratios when we use logit transformation. The odds ratios are intuitive and provide an understanding of the feature effects. On the other hand, the least-squares estimation cannot be used to fit the logistic regression model [32]. For this reason, we need to formulate the maximum likelihood estimation function which maximizes the log-likelihood of the underlying data. Let $\boldsymbol{\beta}$ be the vector of parameters $(\beta_0, \beta_1, \ldots, \beta_p)'$.

$$\begin{aligned} l(\boldsymbol{\beta}) &= \prod_{j=1}^{N} \pi(X_j)^{Y_j}\left[1-\pi(X_j)\right]^{1-Y_j} \\ L(\boldsymbol{\beta}) &= \sum_{j=1}^{N} Y_j \ln(\pi(X_j)) + (1-Y_j)\ln(1-\pi(X_j)) \end{aligned} \tag{4.16}$$

The resulting $\boldsymbol{\beta}$ parameters are interpreted by their influence on the probability of being innovative. A high positive β_p indicates that increase of probability of being innovative; $p(Y = 1|X)$. On the other hand, the negative β parameters negatively affect the probability of being innovative. Since we transformed the raw texts into frequent words and embedding vectors, the logistic regression model enables us to directly find the most positive and negative influencing words.

The logistic regression is fitted with scikit-learn library in Python (version=0.23.2) [43]. The resulting X feature space was large compared to the available data after the document vector transformations. Consequently, a feature selection is advantageous. The lasso regression provides a non-smooth convex function which facilitates feature selection. The $l1$ regularization used and the regularization term added to the loss function as demonstrated in formula 4.17 where λ_1 is the regularization parameter. As expected, the $l1$ regularization will force certain parameters to zero which will automatically select important features from the large input feature space.

$$argmin_{\boldsymbol{\beta}}\big(L(\boldsymbol{\beta}) + \lambda_1||\boldsymbol{\beta}||_1\big) \tag{4.17}$$

4.6.2 Random Forest

In addition to the logistic regression, a random forest is considered for prediction. The random forest is an ensemble model constructed from decision trees. Although it can be used for both regression and classification problems, the seminal paper of random forest noted its strengths on classification problems [6]. The random forest technique uses the bagging strategy to ensemble individual decision trees. As illustrated in figure 4.5, the feature space X is randomly sampled. The classification trees are also built with the random sample with the replacement of the original websites. Finally, the majority voting is taken into account to return a final prediction.

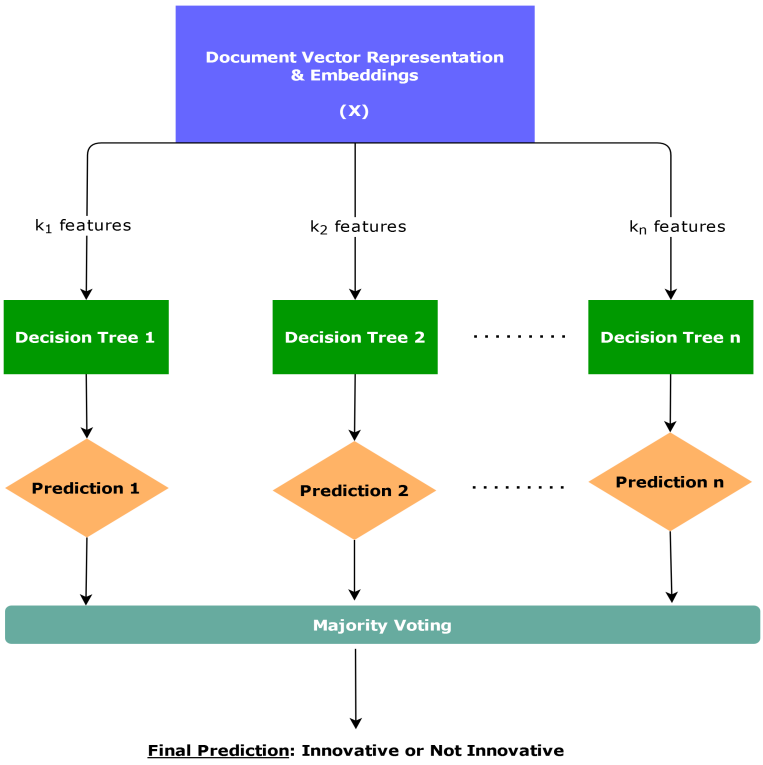

Figure 4.5: Random Forest Representation

Each decision tree is called base learners. Let $h(x)_1 \dots h_n(x)$ be the base learners ensemble together to form a random forest. Then we can define the majority voting as in formula 4.18 for a classification problem [8].

$$argmax_y \sum_{n=1}^{N} I(y = h_n(x)) \tag{4.18}$$

The classification decision trees built with a random sample of business website texts and random features can be tuned with several hyper-parameters. Basically, the decision trees are split into partitions from the root node by using an information gain criterion. Yet, there is no subtle difference between the choices in practice. We used the entropy criterion in our models. The entropy formula is given in formula 4.19 where the p_i is a probability of the arbitrary rule belongs to a given class [44]. The number of trees to grow in a random forest, the minimum number of required samples for internal node split and bootstrap sampling parameters optimized with hyper-parameter tuning.

$$Entropy = \sum_{i}^{c} -p_i \log_2 p_i \tag{4.19}$$

The main advantage of random forest is being able to produce better predictions without over-fitting [6]. The bagging algorithm decreases the random noise of a single decision tree. The methodology can be applied to regression problems with minor changes. The sparsity of our feature space can be handled by the random forest technique as well. These benefits make the random forest technique a popular choice in practical problems. On the other hand, it takes a significant amount of time to build a large random forest compared to the logistic regression and a decision tree. Consequently, the hyper-parameter tuning and application to a real-time prediction scenario need consideration before deployment. Similar to the logistic regression, the scikit-learn library (version=0.23.2) used to build a random forest classifier [43].

4.6.3 Transformers

One additional novel approach to the already existing literature on business website classification is the transformer models. Compared to the previous methodology, the transformers are a relatively new approach that can be used in supervised text classification. The power of the transformers is not only limited to the text classification and they can be used in tasks such as translation and text generation [54].

The pre-trained models allow the transfer learning which can be fine-tuned further with the business website dataset. The transformers are the state-of-art in the current deep learning based text classification since it outperforms convolutional and recurrent neural networks [54]. The main advantage of the transformers is the eligibility of parallelization and overcoming the bottleneck of short-term memory. Unlike the previous NLP methodology that we followed, the transformers have been streamlined to accept raw text input and perform tokenization.

Although the transformers are growing rapidly since 2019, an influential model will be considered in our approach with its robust counterpart; BERT and RoBERTa [13] [35]. These models are trained on a general unlabeled corpus and need to be fine-tuned to achieve good classification performance. In fact, the BERT model is pre-trained with over 3.3 billion word

corpus. In general, the transformer models are criticized for being black-box models where the inner dynamics and the huge parameter space are not readily interpretable. The multiple neural network layers and multi-attention heads increase the complexity of the model. The transformer models are the first transduction models which use self-attention to compute the input and output representations [52]. Given that the attention in transformers can be interpreted as " how much a particular word will be weighted when computing the next representation of the current word." [7].

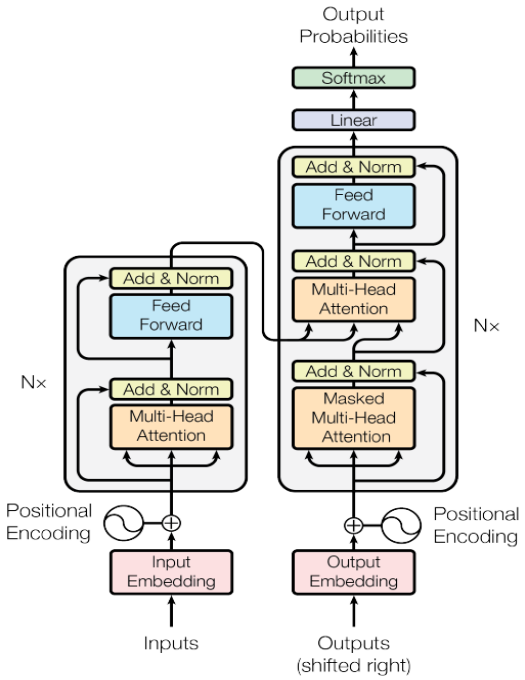

Figure 4.6: The Transformer Model Architecture [52]

In order to fully understand a transformer model, the model architecture given in figure 4.6 by Vaswani and others is helpful [52]. First of all, the transducer model contains an encoder together with a decoder. According to the original notation presented by Vaswani and others, the encoder maps $(x_1, \ldots, x_n)$ to a continuous representation of $(z_1 \ldots, z_n)$ [52]. Given the z vector, the output is generated by the decoder structure as $(y_1, \ldots, y_n)$. The self-attention concept is important for the transformer models since it allows the parallelization of the tasks. The self-attention mechanism investigates the input sequence of $(x_1, \ldots, x_n)$ and internally checks different locations in the input for a better representation of a word. The multi-head attention in the figure 4.6 points to the self-attention mechanism. The left section of the figure 4.6 shows the encoder stack of the transformers. The encoder stack consists of 6 identical layers with two sub-layers within each identical layer [52]. The first sub-layer is the multi-head attention and normalization while the second one is a simple feed-forward neural network. The decoder stack follows a similar layer structure where there are 6 identical layers as in the encoder stack. It is presented on the right side of the figure 4.6. The multi-head attention mechanism is enhanced with an additional sub-layer of masked multi-head attention. The normalization and feed-forward neural network remain in place where the second multi-head

attention is connected to the encoder stack directly.

In more detail, the attention function is responsible for mapping a query with key-value pairs [7]. The query, keys, values and output can be considered as mathematical vectors where the output vector is a weighted sum of the values [52]. Let query vector be $q = (q_1, \ldots, q_n)$, key vector be $k = (k_1, \ldots, k_n)$, value vector be $v = (v_1, \ldots, v_n)$ and lastly the output vector be $o = (o_1, \ldots, o_n)$. The output matrix from them multi-head attention block can be computed as follows where d refers to the dimensionality;

$$Attention_{ij}(q, k) = softmax(\frac{qk^{'}}{\sqrt{d_k}}) = \frac{e^{q_i k_j^{'}}}{\sum_{d=1}^{n} e_i^q k_d^{'}} \tag{4.20}$$

This is a calculation of attention is referred to as softmax-normalized dot product of queries and keys. In other words, the activation function is the softmax function for the multi-head attention sub-layer neurons in figure 4.6. However, the resulting matrix of the attention weights needs to be combined with a weighted sum of the values (v) to return the output vector. The multi-head attention refers to the numerous sets of q, k, v vectors where each set is randomly initialized. The 'Add&Norm' block refers to the addition of the results from these multiple representative sub-spaces and finally normalization.

$$o_i = \sum_{j=1}^{n} Attention_{ij}(q, k) * v_j \tag{4.21}$$

Attention weights can be interpreted as the importance of every tokenized word when producing the succeeding representation of that token [7]. Since we focus on a particular subset of the transformer model (BERT and RoBERTa), the model-specific tokenization needs more consideration. The BERT requires a special tokenization structure such that the beginning of the sequence is marked with [CLS] and the end of the sequence is with [EOS]. Additionally, the [SEP] and [MASK] tokens are used when the raw input text is tokenized. Initially, 15% of the tokens are masked and predicted given the context. This feature of the BERT indicates the masked language model characteristic [13].

On the other hand, there are different meanings of the masking when we investigate the masked multi-head attention sub-layer in figure 4.6. The masked in this sub-layer means a sequence mask where the future information is not exposed to the attention function.

Given the simple softmax classifier $(f(s))$ added on the top of BERT and RoBERTa, we can calculate the probability of innovation. The model would maximize the log probability of the correct label jointly with the attention weights [50]. In the popular implementations of the transformers based on BERT and RoBERTa binary classification, the cross-entropy loss

function (CE) is the default choice.

$$f(s) = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \tag{4.22}$$
$$CE = -\sum_i^C t_i \log(f(s_i))$$

Before the rise of the transformer-based text classifiers, the artificial recurrent neural network was the state-of-art, more specifically long-short term memory (LSTM) [41]. Although there are numerous pre-trained models, the BERT developed by Google has been a top choice since its release [13]. On the other hand, the search for better pre-trained models never ended. We can see the RoBERTa as an improvement to the pre-trained model with longer training times with larger batches [35]. Empirical studies show that the RoBERTa is capable of increasing the predictive capacities of the BERT model [35] [41]. We will investigate the model performances using both of these state-of-art text classification models. To the best of our knowledge, this will be the first application of transformers-based classification on business websites to classify innovative behavior.

The hyper-parameters in transformer models are marginally different from the traditional hyper-parameters due to the nature of the deep learning algorithms. The two most important parameters to optimize are the learning rate and training epochs. In deep learning terminology, the learning rate refers to the rate of adjustment of the model weights in each step. Secondly, the training epochs refer to a complete pass from the training dataset by the algorithm. Since, the transformer-based deep learning algorithms take a considerable amount of time to train (fine-tuning), an extensive search for the hyper-parameters is usually infeasible. Instead, we started with reasonable hyper-parameters and searched for improvements manually. Lastly, the bilingual aspect of our dataset (Dutch and English) hardens the choice of a good pre-trained model. The multilingual pre-trained models exist, but their performance compared to the single language is expected to be more limited.

In the end, the transformers is the state-of-art when we talk about the text classification. The advancements allow for shorter training times with better predictive accuracy. Hence, the deep learning models began to be applied to various datasets. Since the application with business websites in an innovation context is lacking, it is scientifically valuable to evaluate their performance on our unstructured dataset. Given the benefits of these models, they are mostly criticized for being a black-box approach. In a policy-making setting and publicly available official statistics, this aspect of transformers needs additional consideration.

4.6.4 Evaluation Metrics

The binary classification of Flemish businesses implies a 2x2 confusion matrix. The confusion matrix that we can evaluate the model performance is given in figure 4.7. The true positive (TP) and true negative (TN) stand in for correct classifications for innovative and non-innovative businesses respectively. On the other hand, the erroneous predictions included in false positive (FP) and false negative (FN) cells. The false positive is statistically known as Type I error (α) and the false negative is known as Type II error (β).

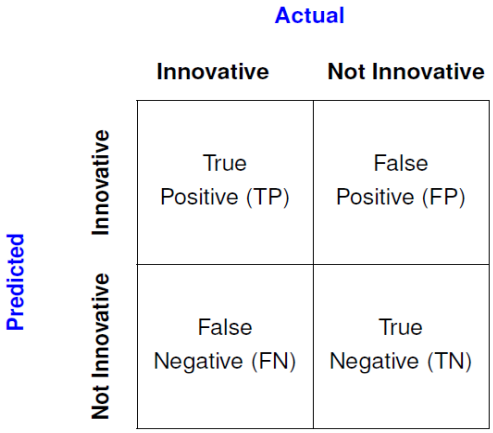

Figure 4.7: Confusion Matrix

There are two different metrics that we will evaluate in our classification model. Firstly, the accuracy score is given in formula 4.20. The accuracy metric is useful to evaluate the predictive power of the underlying model. Yet, the innovative nature of Flemish businesses has a class imbalance problem. Approximately 75% of the Flemish businesses have been labeled as innovative in the input dataset. In other words, one can reach a 0.75 accuracy score just by assigning an innovative label to all companies. Nonetheless, this is not what we would like from a supervised machine learning model. For this reason, the initially high accuracy metric can be a misleading indicator of the model performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.23}$$

A more relevant indicator of model performance is the precision of the binary classification. The precision score refers to the correct predictions among all innovative predictions. Although the precision score is relevant in binary classification, it does not tell all the story. One can adjust the cut-off threshold of predicted probabilities and only select a few certain innovative companies. This would boost the precision score of the model whereas it does not account for a good model. Since the model will end up with high β errors. The recall is a measure of how many innovative businesses are predicted correctly among the actual innovative business set. Both precision and recall equations are presented in formula 4.21.

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN} \tag{4.24}$$

The second evaluation metric is the F1 score where both precision and recall are taken into account. The F1 score is also known as the harmonic mean of precision and recall. A small value in either precision or recall is penalized and the F1 score diminishes. Consequently, it creates a good balance between the two metrics. Hence, the F1 score is preferred over the accuracy score when the problem has a class imbalance.

$$F1\ Score = 2\left[\frac{Precision * Recall}{Precision + Recall}\right] \tag{4.25}$$

Overall, there is no single evaluation metric to rule them all. The choice of evaluation metrics is specific to the underlying problem. There are many other good choices such as Matthews correlation coefficient (MCC) and area under the curve (AUC). Those evaluation metrics have different statistical properties. Nevertheless, the benchmark studies in other countries used the accuracy metric extensively and it constitutes a good comparison setting.

5 Experimental Results

The empirical evidence shows that both logistic regression and random forest algorithms are capable to classify businesses whether innovative or not (see table 5.1). Since there are many possible combinations from text processing steps to the hyper-parameters of the embedding models, it is only possible to search for the limited best options. Hence, the empirical results are investigated in two-folds. Firstly, we present the experimental results for the classification model, embedding model and oversampling combinations. Then, the best combination according to the evaluation metrics will be used to assess various text processing choices similar to a sensitivity analysis.

The text processing steps to get the initial model performances require a clear specification. We have opted to include only Dutch and English texts, remove punctuation and numerical characters. Following the methodology of Statistics Netherlands, we have removed the character sequences of less than three characters [10]. On the other hand, we have altered the text processing methodology that Statistics Netherlands found optimal by excluding the websites that has less than 20 words. The basic motivation to alter the methodology was the observance of the websites with default pointers such as 'expired domain' which usually had more than 10 words but less than 20. Lastly, we have performed stemming as default.

5.1 Classification Results: Test Data

The results table 5.1 provides results from the various model, embedding and oversampling combinations. As default case, all combinations built upon the TF-IDF document vector representation with 300 most frequent words. The selection of the number of frequent words was a result of an experimental trial. Compared to the training dataset, the size of the feature space is reasonable where the higher number of features negatively affects the model performances. Secondly, a result table for the BoW approach is not provided since the BoW document vector representation could not achieve higher evaluation metrics given the above combinations. The TF-IDF document vector representation consistently improved the model results where all else being equal. Considering the methodological superiority of TF-IDF over the BoW, the empirical results are as expected. A hyper-parameter tuning performed using cross-validation of 5 k-folds for both logistic regression and random forests. The independent test dataset was kept aside for an objective assessment.

Classifier	Embeddings	Oversampling	Accuracy	F1 Score
Logistic Regression	Word2Vec	No	0.77	0.47
Logistic Regression	Word2Vec	Yes	0.89	0.78
Logistic Regression	Doc2Vec	No	0.86	0.75
Logistic Regression	Doc2Vec	Yes	0.90	0.80
Random Forest	Word2Vec	No	0.77	0.48
Random Forest	Word2Vec	Yes	0.88	0.78
Random Forest	Doc2Vec	No	0.86	0.75
Random Forest	Doc2Vec	Yes	0.85	0.66

Table 5.1: Classification Results: Test Dataset

The results presented in table 5.1 indicate that both logistic regression and random forests model have low accuracy and F1 score when oversampling is not used. In fact, the models tend to classify all business websites as innovative regardless of the content due to the high class imbalance. Hence, the 77% accuracy creates an illusion of success where the innovative companies have a 75% share in the dataset. Consequently, the F1 scores are more relevant in our context. The oversampling strategy in the training dataset solves the class imbalance problem where the models improve their predictions. The logistic regression with L1 regularization, oversampling and Doc2Vec embeddings resulted in the highest accuracy (0.90) and F1 score (0.80). The random forest achieved similar high results with averaged word embeddings where the final accuracy and F1 score were 0.88 and 0.78 respectively.

In general, both logistic regression and random forest models achieved homogeneous results that were presented by the Statistics Netherlands [10] [9]. The oversampling methodology must be integrated given the unique distribution of the innovative companies in Belgium

compared to the Netherlands. Furthermore, the averaged word embeddings appear to be sufficient for the predictive models where document embeddings provide slight marginal improvement depending on the classification model used. Given the evaluation metrics and the interpretability of the logistic regression model, we utilized the logistic regression model in sensitivity analysis of different text processing steps.

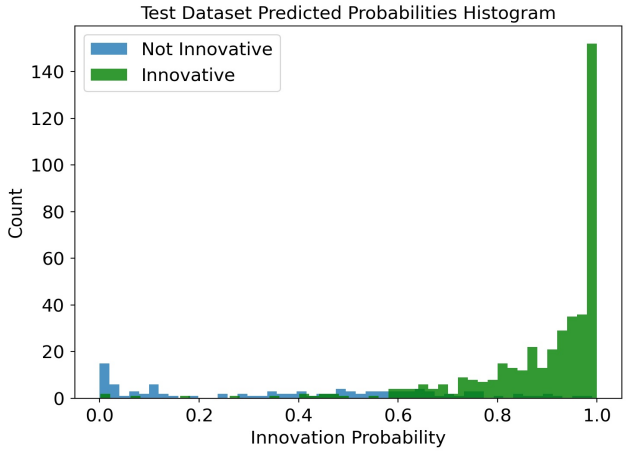

Figure 5.1: Logistic Regression Model: TF-IDF, Doc2Vec & Oversampling

Figure 5.1 shows the predicted probability histogram for the test dataset. As expected, the non-innovative businesses are minority class represented by the blue distribution. As expected from a well-calibrated logistic regression model, the predicted probabilities form a U-shape. Given the class imbalance, the low probabilities are less predicted. The logistic regression model with document embeddings can predict the innovative businesses as innovative with good precision as the figure 5.1 suggests. On the other hand, numerous non-innovative businesses still assigned more than 0.5 probability which is the default threshold. This behavior needs more elaboration which will be discussed in the later section.

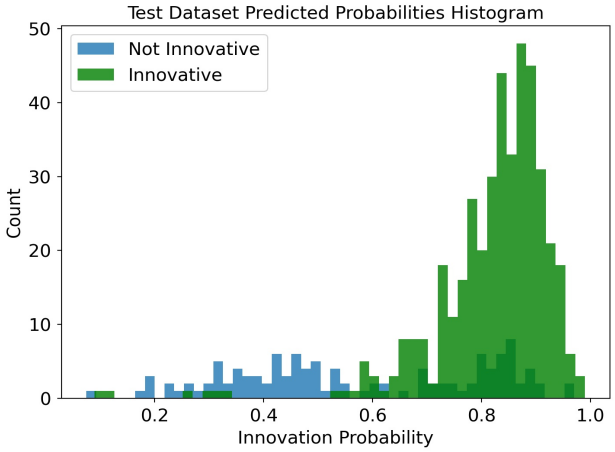

Figure 5.2: Random Forest Model: TF-IDF, Word2Vec & Oversampling

Figure 5.2 demonstrates the predicted probabilities on the test dataset where a random forest classifier is used. Additionally, the document embeddings switched to word embeddings given the better classification results presented in table 5.1. In contrast to the logistic regression model, we can distinguish two clear distributions of innovative and non-innovative businesses. The innovative businesses where true nature is highlighted with green color forms a normal-like distribution with a slight left skew. The predicted probabilities of the innovative businesses appear to be consistent over the threshold which explains the precise results for the category. On the other hand, the not-innovative companies do not form a single normal-like distribution below the threshold. Instead, the predicted probability histogram shows that there are quite a few non-innovative businesses above the threshold and clustered around 0.8 predicted probability. Compared to the misclassified innovative companies, the misclassification of the truly non-innovative businesses is a bigger problem for the classifiers.

5.2 Sensitivity Analysis: Text Pre-processing

The text pre-processing steps are an essential part of any natural language processing pipeline. Although there are countless possible combinations to process raw textual input, the sensitivity analysis of these choices usually paid little attention. We believe that there is no single strategy that can accommodate any raw text input. For this reason, a simple factorial design considering stemming, minimum character sequence, stop word removal and minimum document length has been considered. The stemming, minimum character sequence and minimum document length were suggested by the default methodology presented by Statistics Netherlands [10]. We investigated further the effect of different choices systematically. For the stemming and stop words, language-specific (Dutch or English) tools have been utilized.

Stemming	Char Sequence (min)	Remove Stop Words	Document Length (min)	Accuracy	F1 Score
+	3	+	10	0.88	0.80
+	3	+	20	0.89	0.80
+	3	-	10	0.86	0.76
+	3	-	20	0.87	0.78
+	2	+	10	0.88	0.80
+	2	+	20	0.87	0.80
+	2	-	10	0.86	0.76
+	2	-	20	0.88	0.79
-	3	+	10	0.86	0.78
-	3	+	20	0.87	0.78
-	3	-	10	0.87	0.79
-	3	-	20	0.87	0.78
-	2	+	10	0.87	0.79
-	2	+	20	0.87	0.78
-	2	-	10	0.87	0.79
-	2	-	20	0.88	0.81

Table 5.2: Text Pre-processing Sensitivity Analysis Results

The results from the sensitivity analysis are presented in table 5.2. The logistic regression model with TF-IDF document vector representation and document embeddings is fitted for each combination. Moreover, the accuracy and the F1 Score metrics are averages of 10 different randomly extracted test datasets. Consequently, we have eliminated a possible noise from a single random test dataset. The results in table 5.2 demonstrate that text pre-processing steps have a solid impact on the prediction power of the logistic regression model. In general, the stemming with the combination of a minimum three character sequence outperformed the other combinations. The findings are coherent with the methodology created by Statistics Netherlands [10]. When we consider the stop word removal, the choice appears to be contextually dependent. The combination of stemming and stop word removal surpass the stemming and no stop word removal combination. On the other hand, there is no tangible difference when we do not use stemming operation. This behavior can be explained with the inner mechanics of the TF-IDF formulation where the reduced to stem form counts become comparable to the common stop words. In this scenario, the rareness of the words aforementioned in the methodology section turns into a less useful feature. Hence, the stop word removal is beneficial when we utilize stemming operation alongside. Lastly, there are business websites which hold a tiny amount of textual information. As a fundamental assumption, we need a minimum threshold of word count in order to classify a business website. Overall, the elimination of websites that has less than 20 words produced better results compared to the minimum 10 words threshold. During the sensitivity analysis, we have not considered the lemmatization due to the consistently less accurate predictions compared to the default stemming approach.

The sensitivity analysis demonstrates that the text pre-processing steps are not trivial. In the business website classification, we have found that stemming, minimum 3 character sequence, stop word removal and minimum 20 document length is the winning combination. Depending on the input context, the pre-processing can be fine-tuned further where punctuation and numerical characters can also be considered.

5.3 Transformer Results

The results from the transformer models given in table 5.3. The pre-trained models were acquired from the Hugging Face transformers library [55]. The corpus of combined Dutch and English business website texts has unique characteristics. The pre-trained models are usually trained with a single language and there are few multi-lingual pre-trained models where more than 100 languages are represented. The transformer models also have their own text pre-processing and tokenization pipelines. The default tokenizers for each architecture are used to transform the raw text into a valid input vector. The results indicate that the deep learning approach can improve the prediction results when we consider the F1 scores. As mentioned in the methodically on evaluation metrics, the accuracy of the predictions may be misleading given the context. Although the transformers result in comparable accuracy scores to the traditional classification methods, the F1 scores are consistently higher in transformers. The

main reason for F1 score improvement is a significantly higher recall of non-innovative class. In other words, the transformers retrieve more relevant non-innovative business information from the website texts. The models trained with both 5, 7 and 12 epochs using a learning rate of $2e^{-5}$ and $3e^{-5}$. The training batch size was fixed to 32 and no evaluation between the checkpoints allowed. Due to the long training duration, a complete hyper-parameter optimization was left out of scope.

Architecture	Pre-trained Model	Accuracy	F1 Score
BERT	bert-base-uncased	0.88	0.87
BERT	bert-base-multilingual-uncased	0.88	0.88
BERT	wietsedv/bert-base-dutch-cased	0.91	0.90
RoBERTa	roberta-base	0.88	0.87
RoBERTa	pdelobelle/robbert-v2-dutch-base	0.89	0.89

Table 5.3: Transformers Results: Oversampling

According to the table 5.3, the BERT architecture provided the best results together with the Dutch pre-trained model. Considering the 67.7% representation of the Dutch business website in the corpus, the results are coherent. The base model which has been trained with English text performed worse compared to the multilingual model as well. Yet, even the English base model has better F1 metric and analogous accuracy. The RoBERTa architecture produced nearly identical results when we consider the base models. On the other hand, the Dutch pre-trained RoBERTa could not improve upon the Dutch BERT. Given the additional pre-training of benefits of the RoBERTa architecture and the robustness, the results do not agree with our preliminary expectations.

5.4 Logistic Regression: Parameter Estimates

The logistic regression model is considered to be a white-box predictive technique. Put it differently, it offers a transparent inspection of the parameter estimates. From these parameter estimates, we can interpret the results and how the model infers the business websites. In essence, the logistic regression model was trained with the most frequent words from the TF-IDF and document embeddings. We can focus on the coefficient estimates of the most frequent 20 positive and negative words to better understand the model predictions. These top-20 words given in table 5.4. Taking the innovation concept into account, we can observe the resemblance among the positive words. Development, design, software, technology, research and innovation words are highly relevant in the terminology of innovation. Another interesting aspect of the positive words is that most of the words are English words. The logistic regression model has one more additional binary feature named 'w_language' which is coded as 0 for Dutch websites and 1 for English ones. Given that, this feature does not appear in the top-20 most positive or negative features. Consequently, we can assume that this feature had no significant effect on the prediction where the words and document embeddings provided more predictive information.

Negative Words	Coefficient	Positive Words	Coefficient
offerte	-3.13	applications	3.89
diensten	-1.61	development	2.80
grote	-1.60	health	2.53
industrile	-1.48	design	2.53
transport	-1.41	high	1.91
bezoek	-1.22	software	1.89
ervaring	-1.20	technology	1.88
supply	-1.17	research	1.84
onze	-1.16	jobs	1.73
afspraak	-1.15	time	1.57
medewerkers	-1.12	innovation	1.55
wij	-1.10	systems	1.41
werken	-1.04	one	1.41
welkom	-0.92	view	1.27
fax	-0.91	duurzame	1.13
informatie	-0.87	company	1.12
contact	-0.85	power	1.00
sitemap	-0.81	care	0.90
assortiment	-0.78	products	0.89
email	-0.78	solutions	0.88

Table 5.4: Top-20 Positive and Negative Coefficients - Logistic Regression

The negative words represented more service and traditional industry-oriented terminology. Moreover, we see a more heterogeneous mixture of Dutch and English words. The 'offerte' word has the highest negative coefficient and contextually more likely to present on websites such as insurance providers, banks and retail sellers. None of these sectors are inherently innovative so that it is contextually harmonious. There are two more words among the negative ones that need more attention. The words 'wij' and 'onze' are supposed to be stop words and excluded during the text pre-processing. On the other hand, the NLTK Dutch stop word list turned out to be far from a comprehensive stop word list where these words are not accounted for. In the end, the logistic regression model offers a transparent look behind the prediction dynamics that we can gain valuable information. The top-20 positive and negative word lists indicate that the model actually comprehends the common innovation terminology. Moreover, we can use the words to study which sectors tend to be more likely to be innovative.

5.5 Independent VLAIO Dataset

The innovation agency in Flanders (VLAIO) kindly supplied independent business URLs. These URLs were investigated during their application to VLAIO funds. On the other hand, the VLAIO does not assess and classify companies whether innovative or not. Instead, the agency has various quality labels that can categorize a given business and its fitness for funding.

The lack of true innovation labels guided us to derive one from the VLAIO reports. The annual report from 2019 (Jaarverslagen 2019) provides an exclusive list of supported companies by VLAIO. The suggested indicator for the innovative nature of a given business by VLAIO is to check whether that particular business appears in the list of supported companies.

For the independent test dataset, we have used the best logistic regression model where the text pre-processing steps are kept identical with the training set. The model has been trained with all instances of CIS results and the new accessible VLAIO URLs predicted. In total, we have predicted 2,254 businesses whether innovative or not. The model predicted 1,854 businesses as innovative and 400 as not innovative. Considering the source of the dataset, the innovative percentage above the national average is expected (82.3%).

Given the manual check needed to verify the innovativeness of a business from the annual report, we have opted for checking companies from both the highest and lowest top-100 predicted probabilities. As a result of manual cross-match of business name with annual report showed that there are only 23 innovative businesses among the top-100 businesses with the highest predicted probabilities. Likewise, there are 10 innovative businesses among the top-100 businesses with the lowest predicted probabilities. Although there is a pattern of more innovative businesses among the top-100 businesses with the highest predicted probabilities, the results indicate there is a conceptual problem. Because of that, we have matched the companies from the CIS results and manually checked the VLAIO dataset. Among 200 manually checked companies according to the VLAIO innovation indicator, there were 31 businesses also included in the CIS dataset. The agreement between the true CIS labels and the VLAIO indicator turned out to be only 32.23%. Given the low agreement between the innovation definitions, the results must be interpreted with caution.

6 Discussion

First of all, the basic assumption of classification is that a given business website text encapsulates fundamental innovation concepts. Although the classification task is at the center of our study, the concept of innovation deserves attention at a higher level. The various definitions of the abstract concept of innovation may lead to wrong conclusions and downgrade the model usability. The supervised classification models ultimately take over the definition from the Community Innovation Survey. Hence, a prediction task using an independent test dataset is expected to be classified according to the CIS definition. This feature is useful for Statistics Flanders in order to build low-cost and quality statistics about business innovation. On the other hand, the portability of the prediction models within governmental bodies needs consideration of the rigid innovation concept. The low agreement between the CIS and the VLAIO labels suggests that we need a more unified indicator of innovation at least in the Flemish region.

The measurement of business innovation via CIS throughout the EU states follows a standardized and traditional approach. The use of web scraping and text mining methodologies

was successfully applied in several European countries such as Poland, Germany and Netherlands. In the light of these past studies, the application to the Flemish businesses produced fruitful results and extend the existing methodologies. The traditional featurization through document vector representations and contextual embeddings render satisfactory predictive results on the test dataset. The traditional predictive models can be perceived as white-box models where we can infer from the input features. The transparency aspect of the official statistics has an importance to the statistical authorities. Hence, we need to evaluate models on the interpretability aspect along with their predictive power. In this regard, the deep-learning models (transformers) are black-box models where we can hardly gain information about the inner model dynamics. The hidden layers and the attention weights are innumerable and their individual effect on the outcome is hard to track. Yet, the transformers outperformed the traditional models which use traditional featurization. The detailed classification reports indicate that raise in F1 score is related to the better recall of non-innovative businesses. As the predicted probability histograms suggested (figure 5.1 and 5.2), the traditional algorithms struggle mostly to classify non-innovative businesses. Even though we have utilized an oversampling strategy in training, the mere duplication of the raw text supplied insufficient information to the model. Put it differently, the duplication of the non-innovative business websites successfully solved the class imbalance problem but the duplication did not provide additional useful information for classification. Consequently, the prediction of the non-innovative businesses has still potential to improve given more labeled non-innovative business websites.

In addition to deducting insights from the most positive and negative words from the logistic regression model, we can also mathematically interpret the coefficient estimates. The logistic regression model returns log-odds information. We can transform the default log-odds to odds ratio by taking the exponent of the coefficients. For instance, the odds ratio of most positive word (applications) and most negative word (offerte) are $exp(3.89) = 48.91$ and $exp(-3.13) = 0.04$ respectively. Since the features from the TF-IDF are continuous variables, we can easily interpret the result when we consider all else is held constant. For the most positive word applications, a single unit increase will result in an improvement of odds by a factor of 48.91 to be innovative. Likewise, a single unit increase of TF-IDF weight of most negative words for a business website would decrease the odds of being innovative by a 0.04 multiplicative factor. Considering how relevant the top-20 positive words to the general innovation terminology, the calculated odds ratios are coherent. To conclude our discussion on the model selection, we believe that the white-box models provide useful insights with less predictive power. The transparency and wide comprehension of the traditional models are more useful while the statistical authorities are still exploring the modern tools. At the same time, full deployment and pipeline still benefit from the deep learning models in the near future when the big data application becomes more integrated within the statistical authorities.

Following the empirical results, we can confidently answer our main research question; whether an official statistic can be constructed from the Flemish company websites using web

scraping and text mining methodologies. The predictive models suggest that CIS results can be predicted accurately using web scraping and text mining techniques. On the other hand, there are a couple of limitations. First of all, the Statistics Flanders still working on gathering a comprehensive list of Flemish business information. The official statistic on business innovation should be extracted from a census-like approach to all available Flemish business websites. For this reason, the predictive models form a solid baseline for a quality official statistic on business innovation but a broader application requires additional input data. Secondly, the methodology followed built upon the labeled dataset from the CIS 2019. The supervised aspect of the machine learning techniques depends on these true labels. Considering that the production of an official statistic using the modern methodology should be thought of as an alternative methodology rather than a substitute.

The scalability of the methodology can be achieved with a pipeline prepared exclusively for the Statistics Flanders. The one drawback is that the pipeline assumes the correct URL input for each Flemish business As mentioned earlier, the Flemish authorities are currently working on gathering a comprehensive database of business URLs. Given the URLs, the pipeline is able to harvest website homepage text, pre-process it, train models and even predict independent business websites using a saved model. The Python module prepared is a scalable approach that can be used by Statistics Flanders to build cost-effective, quality and timely official statistics on business innovation (see section 9).

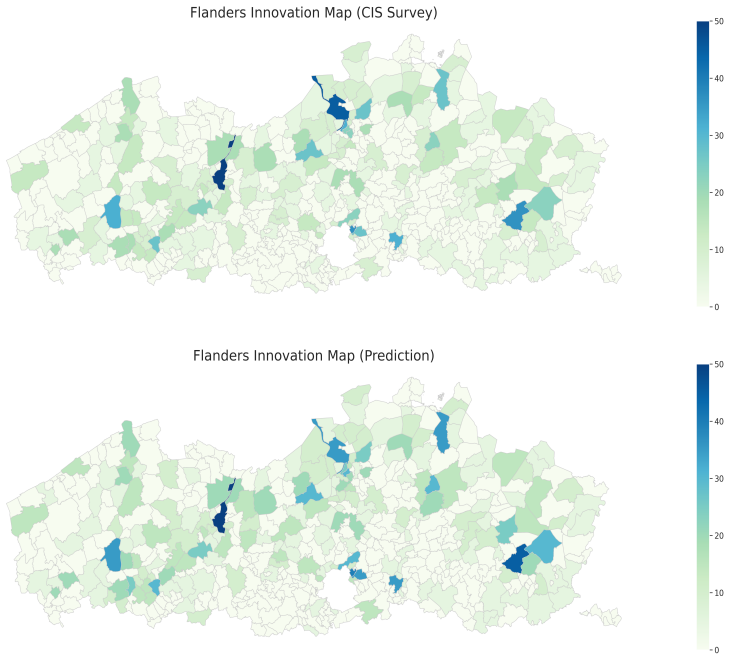

Figure 6.1: Choropleth Map: Count of Innovative Businesses in Municipality Level

A geospatial representation of the innovative businesses in municipality level is given in figure 6.1. The top choropleth map visualizes the count of innovative businesses in the test data set. Consequently, the true labels were counted and colored according to the CIS results. Given the relatively limited size of the test dataset, the map should not be interpreted as a true view of Flemish innovation centers. The choropleth map below in figure 6.1 maps the predicted labels in the same way. Hence, we can compare the results visually and observe abnormal municipalities.

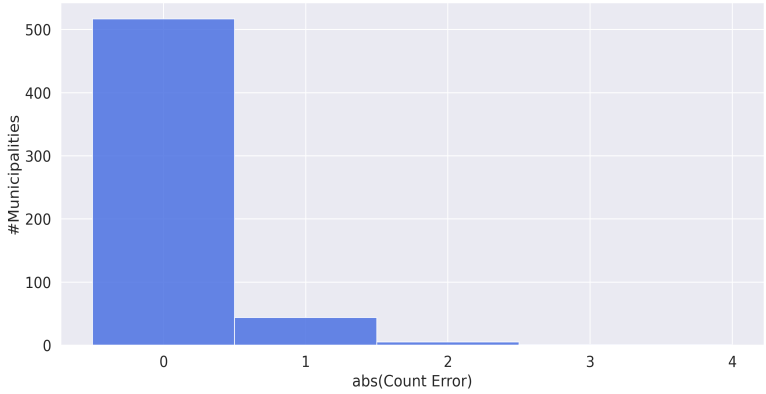

Figure 6.2: Choropleth Map: Absolute Count Error Difference Histogram

Figure 6.2 shows the absolute error histogram from the counts by municipalities. When aggregated on the municipality level, there are a small fraction of count errors for the test dataset. Even though we cannot utilize the choropleth maps to study the geospatial distribution of innovation in the Flemish region due to the limited dataset, the integration forms a baseline for scalability. When the geospatial analysis is applied to all Flemish businesses, we can identify the municipalities that are outlying in the error distribution. Furthermore, a census-like prediction will allow Statistics Flanders to build infographics to disseminate the spatial innovation density to both public and governmental bodies.

7 Conclusion

The aim of the thesis was to construct an complementary methodology to estimate business innovation within the Flemish region. The current official estimates ultimately derived from the Community Innovative Survey (CIS) which uses a primary information source. Given its broad application in all EU states, the statistical authorities experimented with organic and big data techniques in order to reproduce the survey results. In this developing context, we heavily referenced the methodology taken by Statistics Netherlands [9] [10].

Given the unique position of Belgium (see figure 1.1), the classification of Flemish businesses created new challenges in addition to the dynamic web content and multilingual characteristic of Belgium. In order to deal with regional challenges, we have proposed a tailor-made

pipeline to classify business websites whether innovative or not. The pipeline includes every necessary step from the acquisition of the data to the prediction of unseen businesses using saved models.

The traditional pipeline which deals with text pre-processing and manual transformations of raw text input resulted in a satisfactory predictive power. The logistic regression model together with document embeddings and oversampling achieved 0.90 accuracy and 0.80 F1 score. These metrics are coherent with the empirical results published by Statistics Netherlands [10]. Taking into account the socio-economical and linguistic similarities between the two regions, the results are compatible as expected. On the other hand, a new state-of-art methodology is considered besides the traditional pipeline, namely transformers under deep learning. The empirical model results indicate that the transformers significantly increase the recall of non-innovative businesses so that directly affecting the F1 scores. Nonetheless, the deep learning techniques suffer from less interpretability compared to the white-box approaches such as logistic regression. In the context of the official statistics, the accountability and transparency of the estimates need extra consideration along with the predictive power of the models.

In parallel with supervised model exploration, we have also presented a sensitivity analysis for various text pre-processing steps. These preliminary steps are overlooked in the literature of text classification using dynamic website content. Moreover, we have demonstrated of prediction of our best models using an independent VLAIO dataset. Although the overall distribution of binary outcomes appears to be reasonable, an inspection of the top-100 most positive and negative businesses showed that there is a discrepancy between the innovation labels. Since the concept of innovation is quite abstract, the discordant definitions of innovation between the CIS (inno5) and VLAIO followed a low agreement rate. Given that, the results need to be interpreted carefully. Lastly, we have implemented a geospatial integration using choropleth maps that can be scaled when a census-like application will be performed.

Meanwhile, there are certain limitations for the modern methodological estimate. First of all, the supervised learning models still depend on the true label from the CIS. Consequently, modern techniques should only be acknowledged as a complementary methodology rather than a substitute. Secondly, the absence of comprehensive URL domains of all Flemish businesses restricted our application to the businesses that participated in CIS 2019. A full application would require more scalable and robust tools which can accommodate web dynamics and model biases. The training of embeddings can also be affected by the number of websites and their current context. In the end, the supervised models seem to struggle with the classification of non-innovative businesses as the predicted probability histograms indicate (figure 5.1, 5.2).

In the end, this thesis forms a baseline application that can be extended. Along with the replication of results from other statistical authorities, we presented state-of-art text mining

and classification applicants on a similar dataset. In the meantime, there is more theoretical work needed in order to integrate organic and big data sources in official statistics production process.

8 Further Research

The research presented mainly focuses on the baseline application to limited survey data. A more rigorous application to all Flemish businesses is necessary in order to derive a quality official statistic. Further research should give attention to a census-like application whenever an exhaustive list of URLs becomes accessible. On the other hand, the machine learning models trained during the empirical analysis are subject to degradation. Like everything, the time works against the power and stability of the models which need notable attention. For this reason, an investigation of the model degradation for the models trained upon dynamic website texts is crucial. Additionally, a comprehensive application to the entire Flemish region needs more theoretical work where bias assessment is necessary. The identification and analysis of inaccessible websites and missing data may cause a biased estimate of the number of innovative enterprises. Lastly, the deep learning models (transformers) are actively developing. The number of pre-trained models doubled since I have started this thesis. Given the dynamics and rise of deep learning methodologies, a future look into new state-of-art applications will be rewarding.

9 Code

The code that was produced during the application can be openly accessed from following Github repository: *http://www.github.com/nusretipek/Flanders_Innovation*

References

[1] Charu C Aggarwal. *Machine Learning for Text*. Springer International Publishing AG, Cham, 2018. ISBN 9783319735306.

[2] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[4] Sarah E. Boslaugh. Innovation. [Online]. Available from: `https://www.britannica.com/topic/innovation-creativity`, 2016. Encyclopædia Britannica, inc.

[5] Michael Bowers, Dionysios Synodinos, and Victor Sumner. *Pro HTML5 and CSS3 Design Patterns*. Apress L. P, Berkeley, CA, 1 edition, 2011. ISBN 1430237805.

[6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[8] Adele Cutler, D Richard Cutler, and John R Stevens. Random forests. In *Ensemble machine learning*, pages 157–175. Springer, 2012.

[9] Piet J.H. Daas and Jelmer Jansen. Model degradation in web derived text-based models. pages 1–8, 07 2020. doi: 10.4995/CARMA2020.2020.11560.

[10] Piet J.H Daas and Suzanne van der Doef. Detecting innovative companies via their website. *Statistical Journal of the IAOS*, 37, 2020. ISSN 1874-7655.

[11] Piet J.H Daas, Marco J Puts, Bart Buelens, and Paul A.M. van den Hurk. Big data as a source for official statistics. *Journal of official statistics*, 31(2):249–262, 2015. ISSN 2001-7367.

[12] Fred Damerau, Nitin Indurkhya, Sholom M Weiss, and Tong Zhang. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer-Verlag, New York, NY, 1. aufl. edition, 2004. ISBN 0387954333.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Rabiyatou Diouf, Edouard Ngor Sarr, Ousmane Sall, Babiga Birregah, Mamadou Bousso, and Seny Ndiaye Mbaye. Web scraping: State-of-the-art and areas of application. pages 6040–6042. IEEE, 2019. ISBN 1728108586.

[15] Mine Dogucu and Mine Çetinkaya Rundel. Web scraping in the statistics and data science curriculum: Challenges and opportunities. *Journal of statistics education*, ahead-of-print (ahead-of-print):1–11, 2020. ISSN 1069-1898.

[16] Marisa Dziallas and Knut Blind. Innovation indicators throughout the innovation process: An extensive literature analysis. *Technovation*, 80-81:3–29, 2019. ISSN 0166-4972.

[17] EUROSTAT. *Community Innovation Survey: latest results*. 2019. URL `https://ec.europa.eu/eurostat/web/products-eurostat-news/-/DDN-20190312-1`.

[18] Jeffrey EF Friedl. *Mastering regular expressions*. " O'Reilly Media, Inc.", 2006.

[19] Matthew Gentzkow, Bryan Kelly, and Matt Taddy. Text as data. *Journal of economic literature*, 57(3):535–574, 2019. ISSN 0022-0515.

[20] Kranti Vithal Ghag and Ketan Shah. Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 international conference on computer, communication and control (IC4)*, pages 1–6. IEEE, 2015.

[21] Abdullah Gök, Alec Waterworth, and Philip Shapira. Use of web mining in studying innovation. *Scientometrics*, 102(1):653–671, 2014. ISSN 1588-2861.

[22] Abdullah Gök, Alec Waterworth, and Philip Shapira. Use of web mining in studying innovation. *Scientometrics*, 102(1):653–671, 2014. ISSN 1588-2861.

[23] Gustav Haraldsen, Jacqui Jones, Ger Snijkers, and Diane K Willimack. *Designing and conducting business surveys*. Wiley series in survey methodology 568. Wiley, Hoboken (N.J.), 2013. ISBN 9780470903049.

[24] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[25] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56. Citeseer, 2000.

[26] Mark Johnson. How the statistical revolution changes (computational) linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational, Linguistics: Virtuous, Vicious or Vacuous?*, pages 3–11, Athens, Greece, 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W09-0103`.

[27] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[28] J. Katz and Viv Cothey. Web indicators for complex innovation systems. *Research Evaluation*, 14, 01 2006.

[29] Jashanjot Kaur and P Kaur Buttar. A systematic review on stopword removal algorithms. *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng*, 4(4), 2018.

[30] Jan Kinne and David Lenz. Predicting innovative firms using web mining and deep learning. *SSRN Electronic Journal*, 01 2019. doi: 10.2139/ssrn.3321060.

[31] Jan Kinne and Bernd Resch. Generating big spatial data on firm innovation activity from text- mined firm websites. *GI Forum ...*, 1:82–89, 2018. ISSN 2308-1708.

[32] David G Kleinbaum and Mitchel Klein. *Logistic Regression: A Self-Learning Text*. Statistics for Biology and Health. Springer New York, New York, NY, 2002. ISBN 1441917411.

[33] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[34] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[38] Marcin Mirończuk and Jaroslaw Protasiewicz. A diversified classification committee for recognition of innovative internet domains. volume 613, pages 368–383, 04 2016. ISBN 978-3-319-34098-2. doi: 10.1007/978-3-319-34099-929.

[39] Marcin Michał Mirończuk and Jarosław Protasiewicz. A recent overview of the state-of-the-art elements of text classification. *Expert systems with applications*, 106:36–54, 2018. ISSN 0957-4174.

[40] Marcin Michał Mirończuk and Jarosław Protasiewicz. Recognising innovative companies by using a diversified stacked generalisation method for website classification. *Applied intelligence (Dordrecht, Netherlands)*, 50(1):42–60, 2019. ISSN 1573-7497.

[41] Ankit Murarka, Balaji Radhakrishnan, and Sushma Ravichandran. Detection and classification of mental illnesses on social media using roberta. *arXiv preprint arXiv:2011.11226*, 2020.

[42] Eric Nalisnick and Sachin Ravi. Learning the dimensionality of word embeddings. *arXiv preprint arXiv:1511.05392*, 2015.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] Omid Rahmati, Mohammadtaghi Avand, Peiman Yarian, John P Tiefenbacher, Ali Azareh, and Dieu Tien Bui. Assessment of gini, entropy, and ratio based classification trees for groundwater potential modeling and prediction. *Geocarto International*, pages 1–18, 2020.

[45] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[46] Christian Reimsbach-Kounatze. The proliferation of "big data" and implications for official statistics and statistical agencies, 2015. URL `https://www.oecd-ilibrary.org/content/paper/5js7t9wqzvg8-en`.

[47] Constant Rietsch, Catherine Beaudry, and Mikael Heroux-Vaillancourt. Validation of a web mining technique to measure innovation in the canadian nanotechnology-related community. 07 2016. doi: 10.4995/CARMA2016.2016.3140.

[48] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[49] Dipanjan Sarkar. *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing*. Apress L. P, Berkeley, CA, 2 edition, 2019. ISBN 1484243536.

[50] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[51] Seppe vanden Broucke and Bart Baesens. *Practical Web Scraping for Data Science: Best Practices and Examples with Python*. Apress L. P, Berkeley, CA, 2018. ISBN 9781484235812.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[53] Daniel Vorgrimler, Gorja Bartsch, Florian Spengler, and Daniel Kuehnhenrich. Measuring the response burden of official statistical surveys for businesses. *AStA Wirtschafts- und Sozialstatistisches Archiv*, 9(1):59–71, 2015. ISSN 1863-8155.

[54] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference*

on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, 2020.

[55] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[56] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–37, 2008.

[57] Jan Youtie, Diana Hicks, Philip Shapira, and Travis Horsley. Pathways from discovery to commercialisation: using web sources to track small and medium-sized enterprise strategies in emerging nanotechnologies. *Technology Analysis & Strategic Management*, 24 (10):981–995, 2012. ISSN 0953-7325.

[58] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, 1:43–52, 12 2010. doi: 10.1007/s13042-010-0001-0.

[59] Rui Zhao and Kezhi Mao. Fuzzy bag-of-words model for document representation. *IEEE Transactions on Fuzzy Systems*, PP:1–1, 03 2017. doi: 10.1109/TFUZZ.2017.2690222.