FYT Final Report

Business Lead Qualification by Online Information Scraping

RO2

by

Ku Chun Kit

Advised by

Dr. David Rossiter

Submitted in partial fulfillment

of the requirements for COMP 4981H

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2016-2017

Date of submission: April 20, 2017

Table of Contents

Table of Figures

Table of Tables

Abstract

With the development of machine learning algorithms and readily available online information, the automation of predictive business lead qualifications becomes possible. Now with automatic lead qualification, a business owner can train a lead qualifier with legacy sales data and client information automatically scraped online, then submit information concerning a new business prospect to let the qualifier decide how high the chance it is to win a deal. Business owner can save time on information collection and focus on actual selling activities. This paper describes how I developed an automatic business lead qualifier using machine learning and online information scraping.

I first collect real sales data from a partnering company, then classify the companies into two groups depending on whether a deal was made with that company or not. Then the system retrieved company website URL and scraped information from the company websites and social network profile pages. The information was then cleaned up and used to train three models. Predictions generated by models were combined using an algorithm to collectively qualify new business leads.

After testing, the URL guesser achieved around 70% accuracy and the ensemble of models obtained median F1 score of 0.7-0.79 and median area under ROC curve (AUC) of 0.8-0.89.

1 Introduction

1.1 Overview

Lead qualification is a critical process in business-to-business companies. Businesses need to learn about the business lead to decide whether the good or service they are selling meet the need of that lead. Traditionally lead qualification is done manually by asking the business lead their budget, authority, need and timing [1]. It may take several phone calls and email exchanges before the sales person can qualify or disqualify a lead. Sales representatives might also capture additional data from company website and blogs [2]. The actual selling only occurs afterwards. Such process is unproductive because all energy of the sales should focus on selling and making deals instead of information gathering.

With the aid of machine learning from legacy sales data and public information scraped from the web, it becomes possible to integrate predictive lead qualification into sales process on top of the traditional lead qualification process. This thesis demonstrates a methodology to qualify business lead by feeding information about a company from the web into a predictive model trained with legacy sales data.

This thesis extends previous works by incorporating social network feeds/updates scraping to increase the amount of up-to-date content available for machine learning algorithms; and implementing text feature extraction using Word2vec [3].

1.2 Objectives

This thesis leverages and repurposes existing knowledge on web crawling, web scraping, natural text processing and classification algorithm to create a predictive business lead qualifier. I have devised methodologies to scrape and process company information from sales data set into machine readable and learnable form and in turn used them to train a lead qualifier.

This final year thesis has the following objectives:

1. Develop a system that automatically collects and scrape company information from the web given company name or company website URL
2. Combine different techniques and algorithms of natural language processing and classification to train an ensemble of predictive models
3. Extend previous work by incorporating social network scraping and performing text feature extraction using Word2vec
4. Evaluate the prediction by feeding test data and forthcoming new client data to the trained model

1.2.1 Assumptions

This thesis assumes that company information available online are effective material in determining whether a deal would be won or not. There are many data such as the person in charge, company internal strategies and budget planning not available online. In this thesis, they would not be addressed as effective data, but they might be inferred via other means. For example, by the industry category and company size the system inferred the company internal strategies and budget as part of the expert knowledge incorporated into the model.

This thesis also assumes past sales data would reflect the future sales data. The company providing sales data for this thesis is selling one B2B (business-to-business) product and would serve similar clients in the future.

This thesis foresees challenge in identifying the critical factor in making a sale. There is many information online that may be useful for qualifying a lead such as company description, news and company size. Different algorithm and methodologies might be used for different factor given. For example, with numbers given, regression analysis could be performed; for text data, classification algorithm like Random Forest are more suitable instead.

1.3 Literature Survey

After performing a survey of related studies on business lead qualifier, the findings are summarized below.

1.3.1 Previous Works

Despite different algorithms and approaches are used, there are works closely reassembling the intention of this study. The methodologies suggested in these studies could be applied to improve the speed and accuracy of this study.

1.3.1.1 Data Cleanup with Natural Language Processing

Information crawled and scraped from web pages are largely in plain text. Thorleuchter, Van den Poel and Prinzie in their study [4] presented strategies for data collection and pre-processing before they can be used to train model. Before text are converted into term vectors in vector-space model, terms are pre-processed as

described in Figure 1.1.

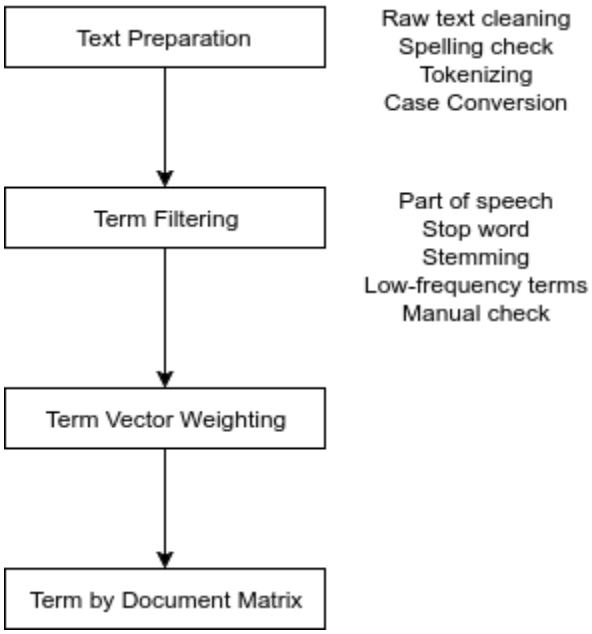


Figure 1.1 Steps taken in [4] to pre-process text data

Non-word terms in the text such as numbers and symbols are removed by categorizing terms into specific syntactic categories namely noun, verbs, adjectives and adverbs). Stop words like articles, conjunctions and prepositions are then removed because they carry little to no information. Stemming and low-frequency terms removal were performed to reduce noise.

A weighed term frequency-inverse document frequency (tf-idf) is then performed by taking the term frequency $tf_{i,j}$ of term i in webpage j times inverse web page frequency idf_i and normalized by their lengths to ensure equal chance of retrieval regardless of their lengths [4].

$$W_{i,j} = \frac{tf_{i,j} \cdot \log(n/df_i)}{\sqrt{\sum_{p=1}^{m} tf^2{}_{i,j_p} \cdot (\log(\frac{n}{df_i}))^2}} \tag{1.1}$$

The approach suggested by [4] enables for transforming data collected from the web to be transformed into machine-learnable form. The process is entirely text-based and

information like some expert knowledge-indicated attributes are completely omitted in the term filtering stage. The algorithm thus is unable to determine, for example, whether there exists a contact form in the webpage. This framework thus lack support for expert knowledge assisted supervised learning [5].

1.3.1.2 Lead Qualification Using Multiple Models

D'Haen and Van den Poel have taken a multi-model approach in generating predictions from both logistics regression and decision tree trained with the same data set [6]. They claim that there is no hypothesis for which model works best and their accuracy may differ depending on the company or industry of interest.

As such, the phase 3 of their three-phase model of with feedback loop incorporates the prediction of the trained model from phase 1 and phase 2 to improve the accuracy of prediction in phase 3.

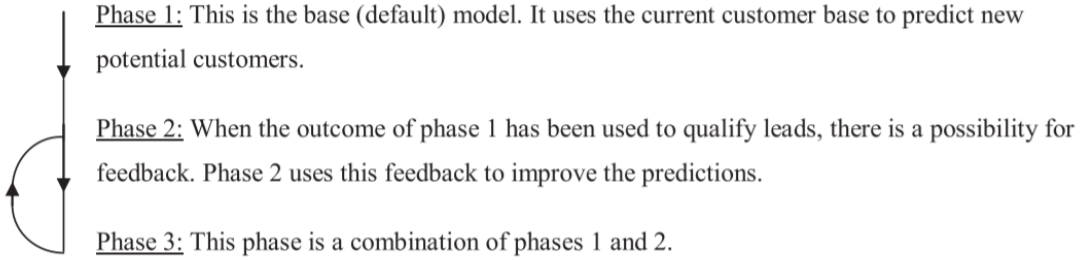


Figure 1.2 Three-phase feedback loop incorporating predictions to train the model [6]

In phase 2 and 3, both logistics regression and decision tree are used to generate a weighted list of results. The basis of comparison is area under an Receiver Operating Characteristic (ROC) curve (AUC), which ranges from 0 to 1.0. An AUC of 0.5 denotes the curve of random guessing, and no classifier should have AUC < 0.5. On the other hand, an AUC of 1.0 represents a classifier of perfect accuracy [7]. By

comparing the AUC of different classifiers, it is D'Haen and Van den Poel uses the following formula to assign weight for the predictions of each of the classifiers in phase 1 and phase 2 respectively.

$$\omega_{Phase\ 2} = (AUC - 0.5) * 2 \quad (1.2)$$

$$\omega_{Phase\ 1} = 1 - \omega_{Phase\ 2} \quad (1.3)$$

Using combination of different classifiers to predict a single outcome would be useful in this thesis because the data mined online are unorganized and largely heterogeneous in industry nature. D'Haen and Van den Poel's approach helps improving the accuracy of our classifiers.

1.3.1.3 Use Web Scraping and Expert Knowledge

D'Haen, Van den Poel, Thorleuchter and Benoit integrated web crawling data and expert knowledge [5], which are several conditions to augment the data set to achieve higher relevance and noise reduction from the multitude of web data source. In the text mining stage, the website structure is analyzed and specific points of interest as defined by expert knowledge are identified. For example, whether a contact form is present in the website or not.

In the work [5] latent semantic analysis (LSA) [8] was used to process and expert knowledge was integrated to achieve an optimal AUC of 0.62, in contrast to that of web data (AUC of 0.526).

In D'Haen and Poel's studies multi-model cross-checking are more effective than single model prediction alone. However, their works mainly use LSA as document processor and logistics regression as classifier, without addressing nor discussing the

efficacy of other algorithms such as Word2vec [9] or Random Forest.

1.3.2 Limitations of Previous Works

Previous works on the topic of business lead qualification have provided groundworks for an accurate lead qualifier. Separate works on preprocessing crawled and scraped data from the web, integration of expert knowledge and model cross-checking are presented and discussed above.

In the case where previous work incorporates web crawling to collect training data set, it failed to recognize the correct website address for given company names [5]. Social media data which are potentially more up-to-date and contain richer information were also failed to be considered part of the training data [5].

However there lacks an integrated study on the efficacy of these approaches combined to process data and train the lead qualifier. Most of the text processing were done using latent semantic analysis without discussion on other approaches such as tf-idf and Word2vec. Logistics regression was the major approach used for machine learning without mention of Random Forest or naïve Bayes.

Therefore, this thesis combines approaches to incorporate additional data from social network profiles using tf-idf, latent semantic analysis and Word2vec to train multiple models to attain greater efficacy in business lead qualification.

2 Methodology

2.1 Design

This final year thesis tests its idea by developing a business lead qualifier that works in five steps as outlined in Figure 2.1.

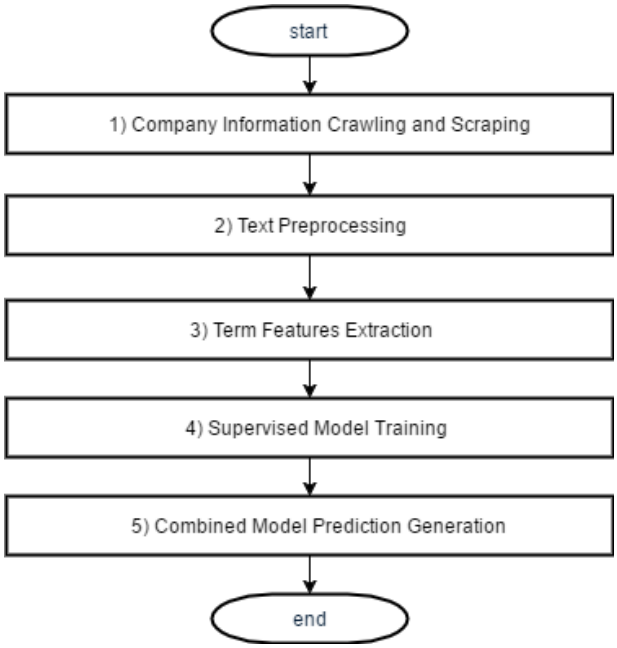


Figure 2.1 Brief overview of the system workflow

The first step is to develop an information scraper to fetch company information online. The second step is to perform text preprocessing on data gathered in step one to reduce noise. The third step involves converting collected online information into term vectors. The forth step trains the models using supervised learning from sales data, term vectors and expert knowledge. The fifth step is to generate prediction a combination of trained models. Detailed view of this workflow can be found on Figure 2.2.

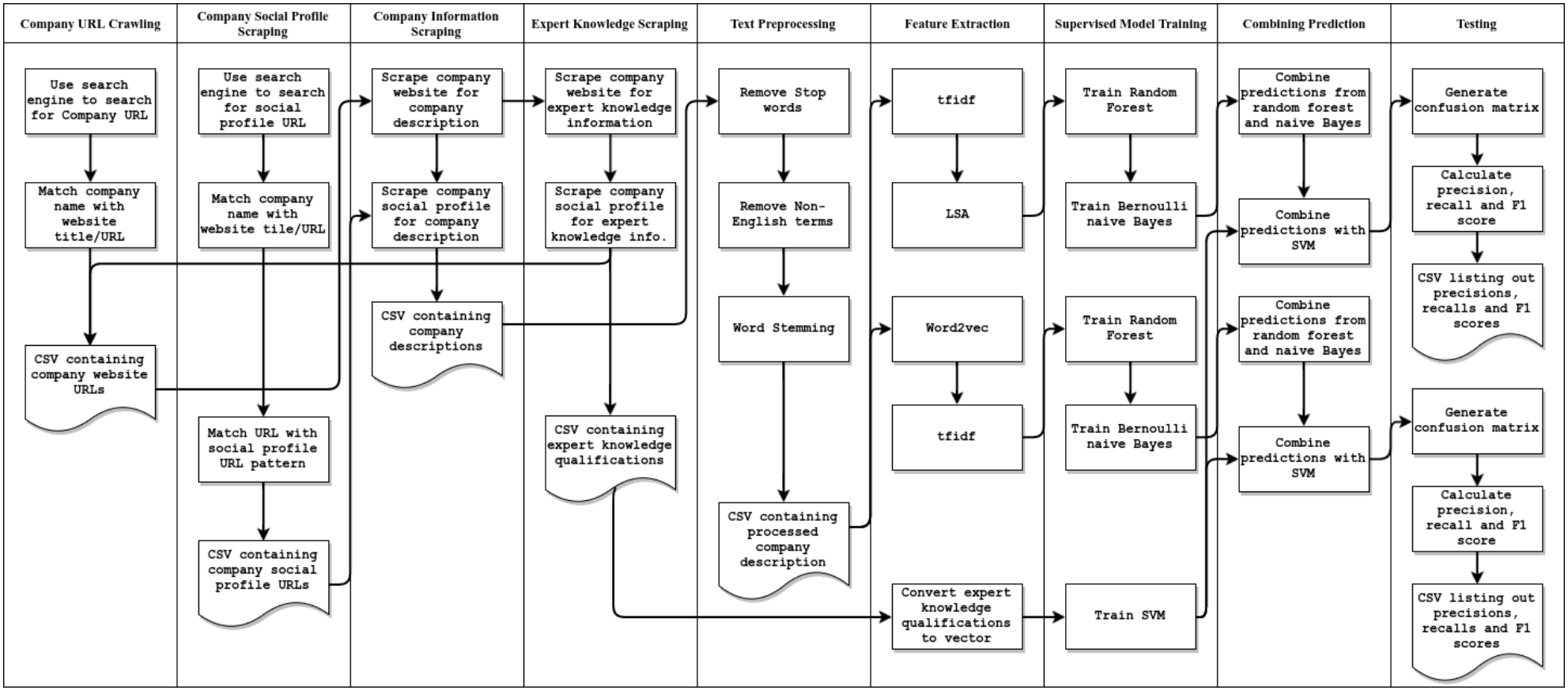


Figure 2.2 Detailed view of the system workflow

2.1.1 Step 1: Company Information Crawling and Scraping

The sales data obtained from the partnering company would contain company name as a starting point. The company names are fed into online search engine to look for their company websites and social network profiles.

Company information is scraped online from company websites and social network profiles using web crawler, scraper and third-party APIs that provides company search service. These fields that are expected to be useful for lead qualification:

- Company description
- Contact information
- Recent activities from company news
- Industry
- Location

The information collected are all in plain text. Additional information such as whether a lead was referred or inbound (client actively contacted the company) are extracted from the sales data.

Previous study [5] suggests that incorporating expert knowledge into model training improves the AUC of the predictors. Taking expert knowledge into account, I surveyed sales people from the partnering company and compiled a list of expert-designated features that will be actively looked for analyzing web data. For example, existences of specific keywords and contact form are organized into a list of true/false attributes concerning a specific company.

2.1.1.1 Company URL Retrieval

Legacy sales data generally lacks or contain incomplete company URL for web scraping. Therefore, company URLs of given company names in the data set need to be retrieved.

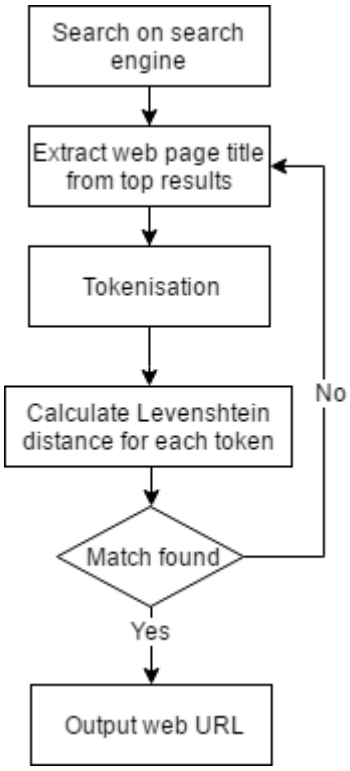


Figure 2.3. Systemic approach of company website URL retrieval

First the web crawler would send search request to search engine and retrieve a list of search results from the site. The crawler assumes that top-ranking results are the most relevant results. The crawler would extract page title from the search results and tokenize them by whitespace after removing common separators like '-' or ','. Likewise, the company name being searched for is tokenized.

For each pair of the tokens *a*, *b* of lengths |*a*| and |*b*|, Levenshtein distance [10] $lev_{a,b}(|a|,|b|)$ is given by:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if\ \min(i,j) = 0; \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \quad (2.1)$$

where $1_{(a_i \neq b_j)}$ represents an indicator function that equals to 0 when $a_i = b_j$ and 1 otherwise; $lev_{a,b}(i,j)$ is the Levenshtein distance between first *i* characters of *a* and the first *j* characters of *b*.

In each web page title, the token pairs with the least Levenshtein distance would be taken to compare that of other web page titles. At the end, the web page title with the least distance would be taken as the company URL. To reinforce and validate the comparison results, the Levenshtein distance between the company name and the URL are compared as well because the company website URL generally contains the company name.

Before outputting the URLs, additional components to the URLs are removed to ensure only domain name and protocol are preserved. For instance, "`https://abc.com/index.html`" would be converted into "`https://abc.com/`".

2.1.1.2 Company Social Profile Retrieval

Nowadays social network profile of a company name along with the social network name would show up as the top-ranking search result in search engines because of the excellent search engine optimization done by the respective social networks.

Therefore, the process of retrieving a company's social profile would be largely the

same as (2.1.1.1 Company URL Retrieval) except that social network names {Facebook, LinkdedIn, Crunchbase} are appended to the search phrase and the resultant URL must be of the same domain as the social networks.

2.1.1.3 Company Information Scraping

Given company URL generated in (2.1.1.1 Company URL Retrieval), the crawler would load the web page and follow the links on the page. Hyperlinks in the page are followed conditionally. For example, if the hyperlinks are leading to the same domain and contain keywords of interest such as ‘details’, ‘company’, ‘service’ and ‘about’. Text from the pages are extracted according to their length and the HTML tags enclosing the text. The scraping result from each page are output as individual rows into a CSV file.

For social profiles from (2.1.1.2 Company Social Profile Retrieval), scraping is performed on both the structured ‘about’ section and the timeline/feed/updates pages respective to the design of the social networks. If the nature of a feed/post/update is web link, the crawler would follow the link and scrape the web page. Since most social profile contains company website URL, the scraping results of this step would be used to reinforce or correct the company URL retrieved from (2.1.1.1 Company URL Retrieval).

Specific expert knowledge-related features would be identified if present. The implementation of an expert knowledge depends on the requirement itself, such as the existence of contact form, specific keywords or even activeness on social networks. Each feature represents one column concerning that company in the resultant CSV

file. The value would be represented by $\{0, 1\}$, with 0 meaning false and 1 meaning true.

2.1.1.4 Using FullContact Company API to Fetch Missing Information

FullContact provides company API [11] which takes company domain name as input, and output company information including company description, location, approximated company size and a list of social profiles links. This API could be used to supplement information unable to gather in (2.1.1.3 Company Information Scraping). Even if (2.1.1.3 Company Information Scraping) performed well and gathered data of interest, the output from FullContact API could still be used to validate the data from (2.1.1.3 Company Information Scraping).

2.1.2 Step 2: Text Preprocessing

Company descriptions acquired from step 1 need to be further processed before they can be used to effectively train a model. To reduce complexity when processing the data, all company descriptions are assumed to be in English. Non-English text would be omitted.

2.1.2.1 Stop Words Removal

Then, stop words such as '*a'* and '*the'* are removed from the text body. There is no one standard stop word list. In this study Boulton's stop word list [12] is used. Stop words need to be removed because they are too common in text that they carry no significant meaning. In numerical statistics such as tf-idf, stop words would be mere noise. It was also suggested that removing stop words helps reducing the dimension of the matrix formed in model training, thus saving memory consumption and speeding up text processing [4].

Table 2.1. Text sample before and after removing non-English terms

Before	But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness.
After	explain mistaken idea denouncing pleasure praising pain born give complete account system, expound actual teachings great explorer truth, master-builder human happiness.
Difference	Terms removed: but, I, must, to, you, how, all, this, of, and, was, will, a, the

2.1.2.2 Non-English Terms Removal

Number and non-alphabetical terms should be removed from the data. Should there remains no or insignificantly short text, that piece of text would be considered ineffective and removed from training data set.

Table 2.2. Text sample before and after removing non-English terms

Before	explain mistaken idea denouncing pleasure praising pain born give complete account system, expound actual teachings great explorer truth, master-builder human happiness.
After	explain mistaken idea denouncing pleasure praising pain born give complete account system expound actual teachings great explorer truth masterbuilder human happiness
Difference	Symbols removed: '-', '.' and ','

2.1.2.3 Word Stemming

After non-English term removal, the company description need to be stemmed. Stemming is the process of truncating suffixes of English words. For example, the word '*class'* and '*classes'* are stemmed into '*class'* while '*classic'* is stemmed as *classic*. There exists Snowball stemmer implementation [13] of the Porter stemming

algorithm [14] would be used.

Table 2.3. Text sample before and after word stemming

Before	explain mistaken idea denouncing pleasure praising pain born give complete account system expound actual teachings great explorer truth masterbuilder human happiness
After	explain *mistaken* idea denounc pleasur prais pain born give complet account system expound actual teach great explor truth masterbuild human happi
Difference	The suffixes of words like 'praising', 'complete', 'explore' are removed without obfuscating the meaning of words. Note that the word 'mistaken' was not truncated to 'mistak' as in 'mistaking' because it would change the meaning.

2.1.3 Step 3: Term Features Extraction

The vector space model is widely used in information retrieval. By turning text into vectors in a vector space, similar documents and relevant terms would cluster together [15]. Two types of term feature extraction techniques namely tf-idf + LSA (count-based method) [8] and Word2vec (a predictive method) [3] are used.

This study is interested in identifying which of the two term feature extraction techniques would be more suitable for transforming company information into learnable format.

2.1.3.1 Vector Generation with tf-idf and Latent Semantic Analysis

The company information collected online is transformed into vector representation first using term frequency-inverse document frequency (tf-idf). Given *N* document*s* $D=\{d_0, d_1 \dots d_N\}$ of vocabulary size *V*, tf-idf would yield a vector of dimension $1\times V$ and there are totally N documents, resulting in a $N\times V$ matrix.

The weight of each $1\times V$ is given by [16]:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{2.2}$$

where

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.3}$$

and

$$idf_i = log \frac{|D|}{|\{j: t_i \in d_j\}|} \tag{2.4}$$

$n_{i,j}$ is the number of occurrence for the term t_i in document d_j; $|\{j: t_i \in d_j\}|$ is the number of documents containing t_i, it would be taken as 1 if the term equals 0.

At this point, the $N\times V$ matrix may be large (when the vocabulary size is huge), noisy (when the input data contain many low tf-idf weight terms) or sparse. To reduce the dimension of the matrix, noise and sparsity, latent semantic analysis (LSA) is used [8].

LSA is performed by using singular value decomposition to dimensionality reduce the tf-idf matrix. In the case of our $N\times V$ matrix A, the SVD [5] takes the form of:

$$A = U\Sigma V^T \tag{2.5}$$

where U is a $N\times N$ orthogonal matrix; V is a $V\times V$ orthogonal matrix.

SVD is applied to the tf-idf matrix [8], in which U contains the document vectors and V the term vectors. The rank-reduced version of A with the largest singular values k is given by:

$$A \approx A_k = U_k \Sigma_k V_k^T \tag{2.6}$$

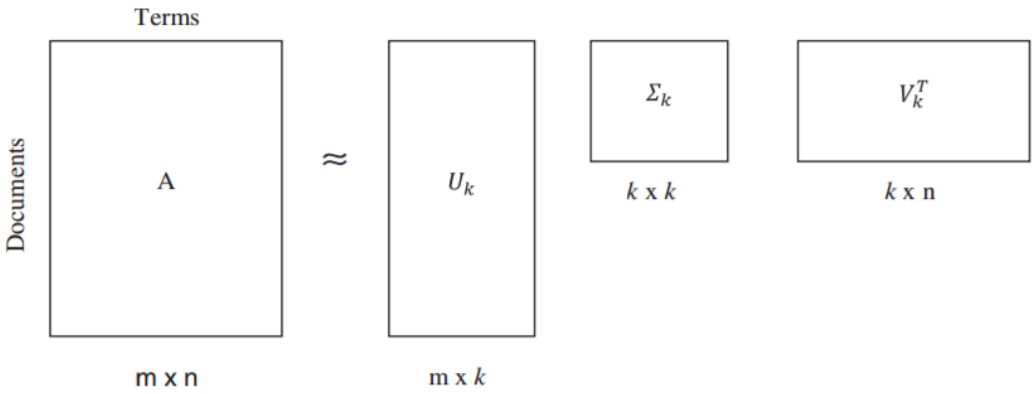


Figure 2.4. Performing SVD for rank lowering on tf-idf matrix [5]

The resultant matrix that has been mapped to a k-dimensional space is known to group similar terms used in similar contexts would have their similar vectors in the reduced-dimensional LSA representation [17]. This clustering behavior is like that of Word2vec despite they are generated from different methods (count-based and predictive).

2.1.3.2 Vector Generation with Word2vec

The Word2vec model proposed by Mikolov et al [3]. is a skip-gram model that learns the relative relationship between words in contrast to the word-relative-to-document statistical measurement in tf-idf. Since Word2vec is relatively new compared to the more popular LSA, in previous studies (1.3.1 Previous Works) nobody has explored the use of Word2vec in term feature extraction and machine learning for lead qualification. It therefore serves the purpose of extending previous works on the topic of lead qualification.

The Word2vec model is implemented in a neural probabilistic language models that maximizes the average log probability [3]:

$$\frac{1}{\mathrm{T}}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log \mathrm{p}(w_{t+j}|w_t) \tag{2.7}$$

where $p(w_{t+j}|w_t)$ is a Softmax function:

$$P(w_O|w_I)=\frac{\exp({v'}_{w_O}^{\mathrm{T}}v_{w_I})}{\sum_{w=1}^{W}\exp({v'}_{w}^{\mathrm{T}}v_{w_I})} \tag{2.8}$$

where W is the number of words in the vocabulary; v_w and v'_w represents the input and output vector through the model respectively.

In contrast to count-based method like LSA, Word2vec explores the relationship between a word and the words surrounding it. This distributional hypothesis ensures that words with similar contexts [18] (for example: 'king is man' and 'bishop is man' suggests both king and bishop are man), hence Word2vec achieves good word representations.

After training the skip-gram model, there should be a set of word embeddings that could be carried on to step 4 for training.

2.1.3.3 Expert Knowledge Feature Vector Formation

In (2.1.1.3), expert knowledge attributes are recorded as a list of $\{0,1\}$ values. They can be converted into $1\times\mathrm{N}$ vectors to represent the company's feature in a N-dimensional space by placing the values $\in\{0,1\}$ into list of integers like:

$$[1,0,1,0,0,0,1]$$

2.1.4 Step 4: Supervised Model Training

2.1.4.1 Model Training with Random Forest and Naïve Bayes

This study intends to combine the predictions from different classifiers (to be discussed further in (2.1.5 Step 5: Combined Model Prediction Generation)), so instead of training only one model, two models are trained using random forest and naïve Bayes separately.

Taking the vectors from Word2vec and tf-idf + LSA from (2.1.3 Step 3: Term Features Extraction), the training data are taken to train two separate random forest [19] classifiers. Several decision trees are generated by tree bagging, that is to generate decision trees from random samples of the training sets with replacement. The ending result of a random forest predictor is the combined results of all decision trees.

A probabilistic approach called naïve Bayes is taken as well. The Bernoulli naïve Bayes is used after comparing it with Gaussian naïve Bayes while Multinomial naïve Bayes is not chosen because it is incompatible with the feature vectors trained. More about their comparison are available in (2.2.4.2 Training Bernoulli Naïve Bayes Model).

Bernoulli naïve Bayes is a popular model for document classification [20]. Bernoulli naïve Bayes classifies a piece of text by the feature term occurrence. A term that exists in the text is given an occurrence value of 1 or otherwise 0. Given a boolean expression x_i for the i′th term in the vocabulary, the possibility that the document is classified as class C_k is [21]:

$$P(x|C_k) = \prod_{i=1}^{n} P_{ki}^{x_i} \cdot (1 - P_{ki})^{(1-x_i)} \tag{2.9}$$

2.1.4.2 Model Training with Expert Knowledge

Expert knowledge, which at this point would be a $1\times N$ vector given N criteria of lead qualification and was determined back in (Company Information Scraping). For example, a vector of 7 expert knowledge criteria would look like:

$$[1, 0, 1, 0, 0, 0, 1]$$

where 1 represents that the expert knowledge criteria are met, and 0 otherwise.

The expert knowledge classifier would be modelled using non-linear support vector machine (SVM) [22]. Since expert knowledge are only supplementary and the vectors are expected to be sparse, it is expected that linear separation would not be sufficient for a good classifier. Therefore, SVM is used to train the model and generate clear cut predictions given expert knowledge.

2.1.5 Step 5: Combined Model Prediction Generation

Although one classifier would work fine by itself, ensemble of classifiers has seen to be improving classification accuracy in supervised machine learning [23].

This study uses what was proposed and used by D'Haen and Van den Poel in [6]. However, instead of AUCs of different classifiers, the F1 score devised in (2.1.4 Step 4: Supervised Model Training) would be calculated and be averaged to respective weights because SVM is not a probabilistic model and SVM on itself does not have per-class probability needed to calculate the AUC. Although its acknowledged that Platt scaling can be used to estimate per-class probability and thus calculate AUC for SVM [24], the more direct F1 score is preferred instead because F1 score [25] captures the eventual prediction correctness only and is not an estimation. It provides for a consistent measurement between each single classifier and the final combined classifier, so F1 score is used. To calculate F1 score of a classifier c:

$$Fscore_c = \frac{2}{\frac{1}{precision_c} + \frac{1}{recall_c}} \tag{2.10}$$

where

$$precision_c = \frac{n_{true\ positive}}{n_{true\ positive} + n_{false\ positive}} \tag{2.11}$$

$$recall_c = \frac{n_{true\ positive}}{n_{true\ positive} + n_{false\ nagative}} \tag{2.12}$$

The weight for each classifier c in a collection of classifiers C is calculated as:

$$w_c = \max(Fscore_c, 0.5) \tag{2.13}$$

Therefore, the prediction jointly computed by text classifiers would be a boolean value given by:

$$prediction_j = \begin{cases} 1, & \sum_k^C w_k p_{kj} > \frac{\sum_{k=1}^C w_k}{2} \\ 0, & \sum_k^C w_k p_{kj} \leq \frac{\sum_{k=1}^C w_k}{2} \end{cases} \tag{2.14}$$

where $p_{kj} \in \{0, 1\}$, is the prediction generated by the classifier k on input feature vector j.

A model may perform worse than random guessing, which has an F1 score of 0.5, by having F1 score less than 0.5. To avoid this faulty model from negatively affecting the output of prediction, its weight w_c would be set to 0.5 to emulate random guessing when $Fscore_c$ is less than 0.5 for classifier c as shown in (2.13).

The above process is repeated for the SVM classifier for expert knowledge, making the combination of three models a two-pass process due to its superior accuracy compared to one-pass process (to be discussed in (2.2.5.1 Comparing One-pass and Two-pass Approaches)). Therefore, the final prediction would be given by:

$$Fscore_{avg} = \frac{\sum_{k=1}^{C} Fscore_k}{2} \tag{2.15}$$

$$p_{combined} = prediction_j \cdot Fscore_{avg} + p_{SVM} \cdot Fscore_{SVM} \tag{2.16}$$

$$\text{final prediction}_j = \begin{cases} 1, & p_{combined} > \dfrac{Fscore_{avg} + Fscore_{SVM}}{2} \\ 0, & p_{combined} \leq \dfrac{Fscore_{avg} + Fscore_{SVM}}{2} \end{cases} \tag{2.17}$$

where p_{SVM} and $Fscore_{SVM}$ are the prediction and F1 score of the SVM classifier given expert knowledge vector for the company having feature vector j.

To provide an overview of the process, the pseudocode of combining model predictions is available below.

```
SET thresholdSum to 0.0
SET countClassifier to 0.0
SET weightedPredictionPassOne to 0.0
FOR each fScore, name of classifiers // except SVM
    SET thresholdSum to thresholdSum + fScore
    CALL findClassifierPrediction with name, sample RETURNING prediction
    INCREMENT countClassifier
    SET weightedPredictionPassOne to weightedPredictionPassOne + fScore * prediction
END FOR

SET thresholdPassOne to thresholdSum / countClassifier
IF weightedPredictionPassOne > thresholdPassOne THEN
    SET passOnePrediction to 1
ELSE
    SET passOnePrediction to 0
END IF

SET thresholdPassTwo to (thresholdPassOne + fScoreOfSVM) / 2.0
CALL findClassifierPrediction with name=SVM, sample RETURNING prediction
SET weightedPredictionPassTwo to passOnePrediction * thresholdPassOne + fScoreOfSVM
* prediction
IF weightedPredictionPassTwo > thresholdPassTwo THEN
    SET passTwoPrediction to 1
ELSE
    SET passTwoPrediction to 0
END IF

RETURN passTwoPrediction
```

2.2 Implementation

This thesis requires a reliable data set from a real B2B company to effectively train a practical model. I partnered with EventXtra Limited to use their sales data from February 2014 to February 2017 under non-disclosure agreement. If a deal is made

with a client, that client's company is labelled as 'qualified', otherwise the company is labelled 'disqualified'.

2.2.1 Developing Company Information Crawler and Scraper

2.2.1.1 Company Website URL and Social Profile Retrieval

A company social profile retriever has been developed. It was implemented in Python using the web scraping library Scrapy. It sends HTTP requests to search engine, parse the response with an HTML parser, and extracted key values using regular expression. The search is done with search engine Google and DuckDuckGo with the URL fabrication rules in the following tables.

Table 2.4. Search engines and URL fabrication for company website retrieval

Search Engine	URL Template
Google	https://www.google.com.hk/search?q={company name}
DuckDuckGo	https://duckduckgo.com/html/?q={company name}

Table 2.5. Search engines and URL fabrication for company social profile retrieval

Search Engine	URL Template
Google	https://www.google.com.hk/search?q={company name + {facebook|crunchbase|linkedin}}
DuckDuckGo	https://duckduckgo.com/html/?q={company name + {facebook|crunchbase|linkedin}}

Originally, only Google was used until Google began returning HTTP 400 Bad Request Error when being accessed with a web scraper. After switching to DuckDuckGo, HTTP 403 Forbidden Error were returned during scraping. In fact, DuckDuckGo's rate-limiting algorithm decided to block the IP being used to access it.

Both Scrapy and web browser were unable to access DuckDuckGo while the IP was being blocked.

To circumvent DuckDuckGo's rate-limiting algorithm, a throttling algorithm was used to calculate the delay between each HTTP request sent to DuckDuckGo according to the latency of the previous requests [26].

$$\text{delay} = \max\left(\frac{1}{2} \cdot \left(\frac{latency}{N} + previous\ \ delay\right), previous\ \ delay\right) \quad (2.18)$$

where N equals the maximum concurrent connections allowed in the setting.

After employing the throttling algorithm and setting the minimum delay to two seconds, DuckDuckGo never returned any HTTP 403 Forbidden Error and the web crawling has been successful.

To ensure the company website and social network profile URL are correct, the sytem calculated the Levenshtein distance of the URLs and their web page titles with the company name. From a list of top-ranking search results, the one with the least Levenshtein distance will be taken.

After several trails, I created a list of blacklisted domains that contains the company name but are information sites listing recruitment or general information of many companies instead of the company website itself. These domains will never be taken as the company website even though the Levenshtein distance may be short. For social network profiles, they generally have a fixed format in URL, so it was possible to achieve a high accuracy on social network profiles by checking the URL format in

addition to Levenshtein distance.

(Appendix D – Sample Company Descriptions Retrieved) shows some links retrieved by looking up the company name and (Appendix E – Sample Social Profile URLs Retrieved) shows some links from searching by company name + social network name on search engine DuckDuckGo.

2.2.1.2 Using FullContact Company API to Fetch Missing Information

After registering for FullContact Company API, I found that their Free plan only allows for 250 company look up per month. The next level in FullContact service plan allows for 24,000 look up per month but the price is prohibitive at $99USD/month.

As such, FullContact Company API was not used for getting company information processing as (2.2.1.1 Company Website URL and Social Profile Retrieval) and (2.2.1.3 Company Information Scraping) are sufficient in acquiring necessary company information.

2.2.1.3 Company Information Scraping

A spider program has been implemented using the Python library Scrapy [27]. It was responsible for scraping the company websites using URLs from both (2.2.1.1 Company Website URL and Social Profile Retrieval) and (2.2.1.4.1 Match Key Information from Social Network Profile).

The spider program first scrapes the frontpage of the company website, looking for hyperlinks with keywords such as ‘details’, ‘company’, ‘service’ and ‘about’ in both the hyperlink title and URL. After that, the spider would follow the links and extract

company descriptions using the Python library called Goose [28]. Goose is an HTML content and article content extractor designed to scrape a website and extract the main text of the page, which is our primary interest in a page.

After scraping, the spider would output a CSV file with column one containing website URL and column two containing company description.

2.2.1.4 Expert Knowledge Scraping

To qualify a company with expert knowledge, the partnering company EventXtra is asked to provide a list of criteria designed by experts in the field. In the context of this thesis, the experts are sales representatives from EventXtra. They provided expert knowledge criteria in form of some questions concerning several pieces of information concerning the companies that can be scraped online. They are:

- Whether the company locates in Hong Kong, Singapore, Taiwan or United States
- Whether the company size is larger than 1000 employees
- Whether the company is in the industries listed in (Appendix F – Qualified Industries According to Legacy Sales data)
- Whether the company has public contact information
- Whether the company organizes events

Expert knowledge scraping has been done in two ways:

1. Match key information from social network profile, retrieving:
    - Company Website (to supplement and validate data from (2.2.1.1 Company Website URL and Social Profile Retrieval))
    - Company description (to supplement and validate data from (2.2.1.3 Company Information Scraping))
    - Headquarter location
    - Company size
    - Industry
2. Look for specific keywords from the site, retrieving:
    - Contact
    - Whether the company organizes events

2.2.1.4.1 Match Key Information from Social Network Profile

Figure 2.4 shows the fields being extracted from a LinkedIn company profile page from accessing an URL retrieved in (2.2.1.1 Company Website URL and Social Profile Retrieval). The textual information enclosed in the blue rectangles are the information extracted by a spider program using Scrapy.

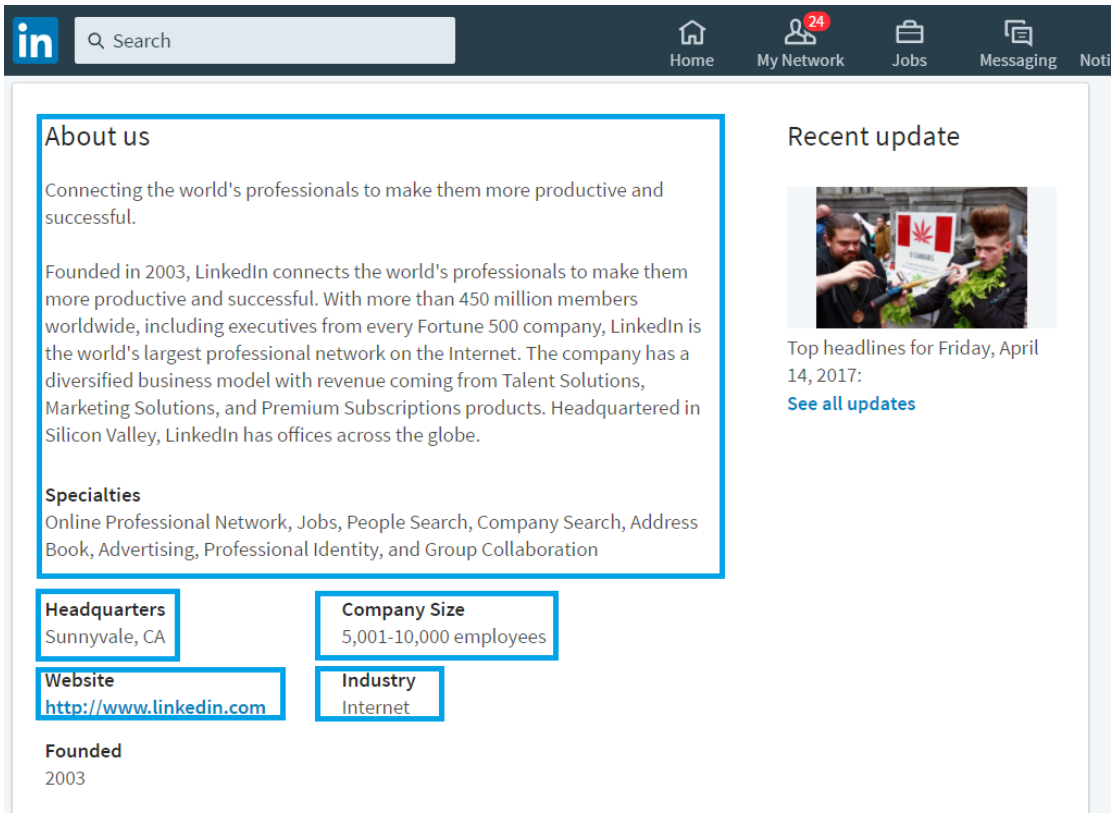


Figure 2.5 A LinkedIn company profile page (after layout update on March 2017)

The company website retrieved was also used to validate or supplement missing company websites URL from (2.2.1.1 Company Website URL and Social Profile Retrieval). Similarly, the company description retrieved was used as output in (2.2.1.3 Company Information Scraping).

Before the spider can access the HTML body of a company profile page, it programmatically logon to LinkedIn with credentials of a valid user account. This was done by sending a HTTP POST requests reassembling a legitimate login request as if a real human was logging in using a web browser. Then, by looking up specific HTML node by its node ID using regular expression, the textual information was extracted.

However, since March 2017 LinkedIn has updated the page structure and layout such

that on HTML GET request, the returned HTML body did not contain the required information directly as HTML nodes. Instead, the information was enclosed in <code> nodes as JSON. The spider program had to be modified heavily to accommodate for this update from LinkedIn, the detailed description of the new page structure and the approach taken to tackle this change can be found in (Appendix G – Detailed Description of LinkedIn Scraping).

The throttling algorithm used in (2.2.1.1 Company Website URL and Social Profile Retrieval) was also used when scraping LinkedIn to avoid the account or IP being banned by LinkedIn. With throttling, the event of LinkedIn banning the account or IP never happened.

2.2.1.4.2 Look for specific keywords from the site

As defined as two expert knowledge qualifiers above, a company is qualified by the existence of contact section and event information on its company website. To retrieve these information, a spider program was developed to access the company website. The URL was either fetched from (2.2.1.1 Company Website URL and Social Profile Retrieval) or retrieved by scraping the LinkedIn profile page in (2.2.1.4.1 Match Key Information from Social Network Profile). The spider program follows hyperlinks pointing to the same domain and searches for keywords namely ‘contact’ and ‘event’ from the page headers and hyperlink titles. Once such keyword has been found, the company becomes qualified as ‘having contact information’ or ‘organizing event’.

2.2.2 Text Preprocessing

2.2.2.1 Stop Words Removal

Boulton's stop word list [12] has been used to remove stop words from the text corpus. A Python script would search through the text and remove any word that is on the stop word list.

2.2.2.2 Removing non-English terms

A Python script has been developed to remove non-English terms from a string. It utilizes the 're' regular expression module to look for non-English, non-space characters and substitute that substring with empty string.

2.2.2.3 Word Stemming

A Python script has been developed using the Snowball stemmer implementation from Python library 'ntlk' to perform word stemming on the text corpus.

2.2.3 Term Feature Extraction

Two separate approaches have been used to extract term features of company descriptions retrieved from (2.2.1.3 Company Information Scraping) and (2.2.1.4 Expert Knowledge Scraping), which was in turn processed in (2.2 Implementation). The first one was a tf-idf followed by LSA. The second one is Word2vec.

To create vectors for incorporating expert knowledge, the information retrieved from (2.2.1.4 Expert Knowledge Scraping) was used to check against the expert-designated criteria.

2.2.3.1 Tf-idf + LSA Approach

A Python script was created to read the information preprocessed in (2.2.2 Text Preprocessing). The data was separated into two sets randomly, 70% of them are training data and 30% are test data.

Then the script instantiates a `TfidfVectorizer` provided by `scikit-learn` library and feed the training data set so it can learn the vocabulary and its idf. Using the same `TfidfVectorizer` instance, both the training and test data were transformed into document-term matrix.

Latent semantic analysis was then performed with `TruncatedSVD` class provided by `scikit-learn`. The desired dimensionality of the output data was specified in the `n_components` parameter of the class constructor and the recommended value for LSA is 100 according to `scikit-learn` documentation [29]. To find the optimal parameter value, LSA was performed with 2, 3, 5, 10, 20, 40, 80, 100, 200, 300, 400, 500 and 600 `n_components` value. According to Figure 2.5, 100 was found to be the optimal value. With other factors being equal, the average F1 score over 10 iterations of the same `n_components` value is at its maximum when `n_components` equals to 100. Although the F1 score is comparable to F1 scores when `n_components` equals to 5 or 400, they were dismissed with the recommendation from [29] was taken into account.

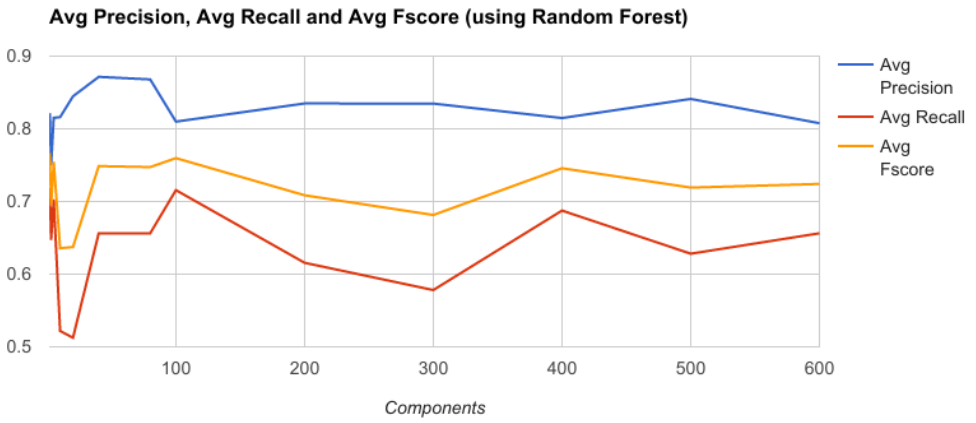


Figure 2.6 Model obtains maximum F1 scores when output dimensionality is 5, 100 or 400

Despite the Random Forest model in (2.2.4 Train Model Using Crawled Data) works best when `n_components` is 100, it was noticed that the optimal `n_components` value for Bernoulli naïve Bayes is 200, according to Figure 2.6. With `n_components` set to 100 for LSA used to feed into Random Forest model, LSA was performed with 2, 3, 5, 10, 20, 40, 80, 100, 200, 300, 400, 500 and 600 `n_components` value. The resultant matrix was exclusively used to train the Bernoulli naïve Bayes model. The ensemble of models (Random Forest, Bernoulli naïve Bayes and SVM with expert knowledge data) and results of 10 iterations for each `n_components` value showed that the F1 score is at its maximum when `n_components` equals to 10 or 20.

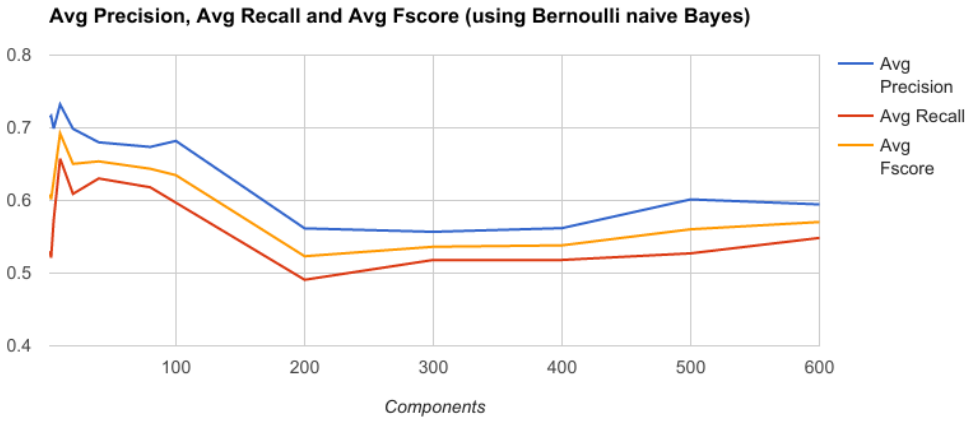


Figure 2.7 Model obtains maximum F1 scores when output dimensionality is 10 or 20 for naïve Bayes

Therefore, two separate `TruncatedSVD` instances was used with one having

`n_components=100` generating matrix used to train and test Random Forest model and one `n_components=20` for training and testing Bernoulli naïve Bayes model.

2.2.3.2 Word2vec Approach

As in (2.2.3.1 Tf-idf + LSA Approach), the data are separated randomly into training data and testing data in 3:7 ratio.

`Word2Vec` module from natural language processing library `gensim` in Python was used. `Word2Vec` was given the training data and used them as the corpus to generate word embeddings. A `TfidfVectorizer` was used in combination with the generated word embeddings to normalize the vectors according the term frequency of each vocabulary by their inverse document frequency in the data.

2.2.3.3 Expert Knowledge Feature Vector Formation

With information retrieved in (2.2.1.4 Expert Knowledge Scraping), the information was used to check against the criteria outlined in (2.2.1.4 Expert Knowledge Scraping) to output an CSV file containing the company names and each criterion with values of either ‘1’ or ‘0’. This CSV file would be carried forward to (2.2.4 Train Model Using Crawled Data) for model training using SVM.

2.2.4 Train Model Using Crawled Data

2.2.4.1 Training Random Forest Model

Random Forest model was trained using `RandomForestClassifier` from `scikit-learn`. `RandomForestClassifier` lets user configure the number of decision trees in the forest with `n_estimators` parameter.

To find out the optimal `n_estimators`, experiments were performed on different `n_estimators` values, with increment by 10 each step from 10 to 590. To decide the optimal `n_estimators` value, F1 score of the trained model and its out-of-bag (OOB) error were calculated. Out-of-bag error is a mean of prediction errors given a sample x_i for trees that did not take a sample x_i during their bootstrap aggregation. Both measures showed mixed results for any estimator number.

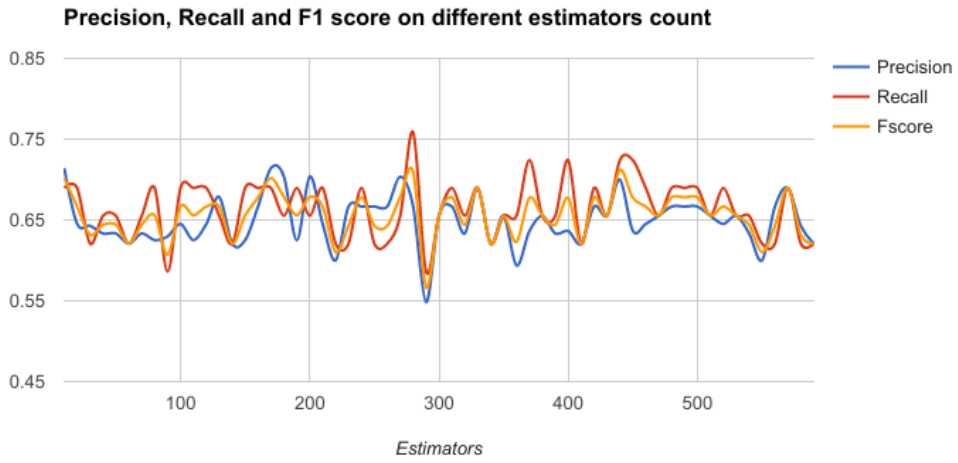


Figure 2.8 Precision, recall and F1 score of random forest model on different numbers of estimators

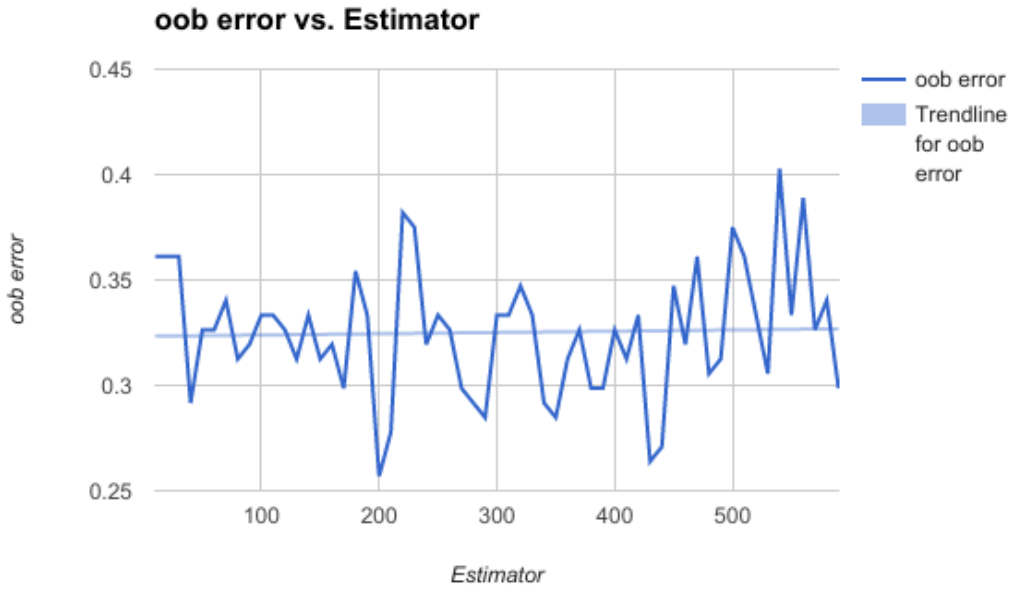


Figure 2.9 Out-of-bag (OOB) error of random forest model on different numbers of estimators

According to Figure 2.7 and 2.8, there was no indicative optimal value of `n_estimators`. There was no visible trend concerning precision, recall, F1 score and oob error on different number of estimators, so `n_estimators` was simply taken as 200.

The random forest model was trained on feature vectors processed by LSA with 100 components according to the analysis in (2.2.3.1 Tf-idf + LSA Approach). Training data was used to train the random forest model and test data was used to calculate the confusion matrix, precision, recall and F1 score. The same has been done on vectors generated from Word2vec in (2.2.3.2 Word2vec Approach), their comparisons will be further discussed in (2.3 Testing).

Table 2.6 Normalized confusion matrix of a trained random forest

		Predicted	
		Positive	Negative
Actual	Positive	0.7273	0.2727
	Negative	0.2333	0.7667

Table 2.7 Precision, recall, F1 score of a trained random forest

Precision	0.7188
Recall	0.7667
F1 score	0.7419

2.2.4.2 Training Bernoulli Naïve Bayes Model

There are three naïve Bayes modules provided by `scikit-learn`: `GaussianNB`, `BernoulliNB` and `MultinomialNB`. Since `MultinomialNB` is suitable for

classification with discrete features, it requires integer feature counts [30] and is incompatible with the weighted normalized vectors obtained from (2.2.3.1 Tf-idf + LSA Approach) and (2.2.3.2 Word2vec Approach), which contains fractions instead of integers, so `MultinomialNB` was not used.

Despite both `GaussianNB` and `BernoulliNB` are suitable machine learning algorithm for this task, `BernoulliNB` was chosen at the end because after training the both models using the same set of data 200 times, `BernoulliNB` outperforms `GaussianNB` with higher median F1 score as shown in Figure 2.7 and 2.8.

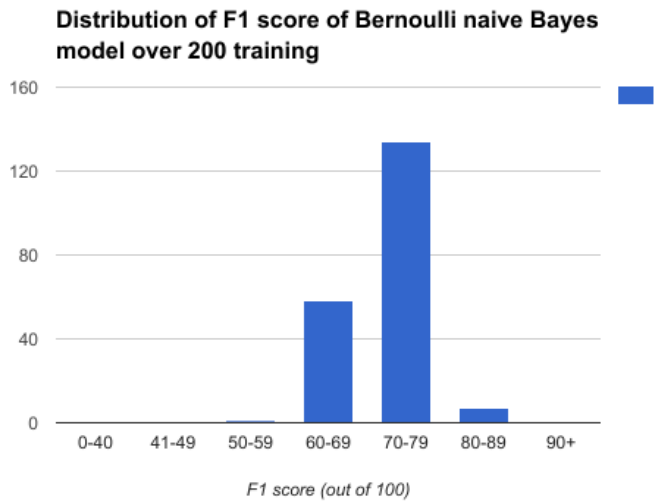


Figure 2.10 With Bernoulli naïve Bayes, the median F1 score is around 70-79 (out of 100)

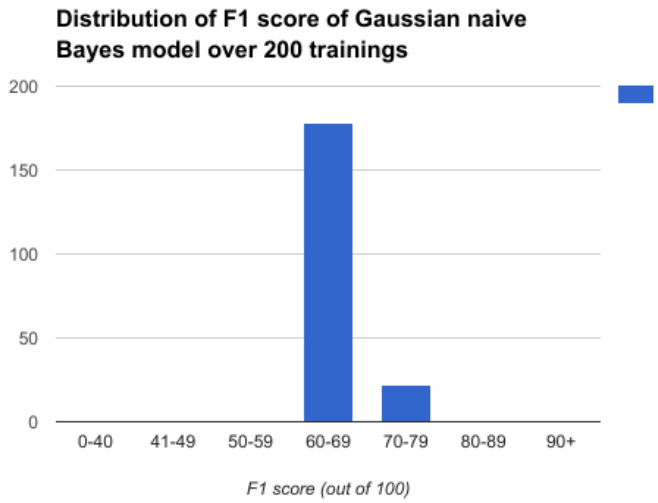


Figure 2.11 With Gaussian naïve Bayes, the median F1 score is around 60-69 (out of 100)

The Bernoulli naïve Bayes model will use feature vectors processed by LSA with 20 components instead of 100 according to the analysis in (2.2.3.1 Tf-idf + LSA Approach). Training data was used to train the Bernoulli naïve Bayes model and test data was used to calculate the confusion matrix, precision, recall and F1 score. The same has been done on vectors generated from Word2vec in (2.2.3.2 Word2vec Approach), their comparisons will be further discussed in (2.3 Testing).

Table 2.8 Normalized confusion matrix of a trained Bernoulli naïve Bayes model

		Predicted	
		Positive	Negative
Actual	Positive	0.8	0.2
	Negative	0.2424	0.7576

Table 2.9 Precision, recall, F1 score of a trained Bernoulli naïve Bayes model

Precision	0.8065
Recall	0.7576
F1 score	0.7812

2.2.4.3 Training Expert Knowledge Model

Expert knowledge model was trained on different sets of qualified and disqualified data compared to the random forest model and Bernoulli naïve Bayes model because it used expert knowledge vectors generated in (2.2.3.3 Expert Knowledge Feature Vector Formation) instead of text vectors from (2.2.3.1 Tf-idf + LSA Approach) and (2.2.3.2 Word2vec Approach). The expert knowledge data was separated in a ratio of 7:3, with 70% being training data and 30% being testing data.

The `svm.SVC` module from `scikit-learn` was used to build a SVM model. To optimize the SVM, different kernel options including linear kernel, radial basis function kernel and polynomial kernel up to 9 degrees. Result showed that linear, radio basis function and polynomial kernel (1 degree) performed on par while polynomial kernel accuracy degraded as the degree was increased. Therefore, radio basis function kernel was chosen.

The training data was fed into an `svm.SVC` instance and the trained model was used to predict outputs from the testing data to calculate the confusion matrix, precision, recall and F1 score, as shown in Table 2.6 and Table 2.7.

Table 2.10 Normalized confusion matrix of a trained SVM

		Predicted	
		Positive	Negative
Actual	Positive	0.9714	0.0286
	Negative	0.3636	0.6364

Table 2.11 Precision, recall, F1 score of a trained SVM

Precision	0.8750
Recall	0.6364
F1 score	0.7368

2.2.5 Generating Predictions from Models Combined

During model training phase, F1 score of each model was taken for calculating their weights when combining the predictions from each model. For instance, if the 3 models have the following F1 score in Table 2.12:

Table 2.12 F1 scores and predictions by random forest, Bernoulli naïve Bayes and SVM respectively

	Random forest	*Bernoulli naïve Bayes*	*SVM*
F1 score	0.7353	0.6667	0.7391
Prediction	1 (positive)	0 (negative)	1 (positive)

2.2.5.1 Comparing One-pass and Two-pass Approaches

Prediction has been done with 2 passes, where the first pass aggregates the prediction from random forest and Bernoulli naïve Bayes model. The first-pass result was then passed on to the second pass to combine with prediction from SVM as described in (2.1.5 Step 5: Combined Model Prediction Generation). An experiment has been done on comparing one-pass and two-pass approach and it has been found that when the F1 score of SVM is higher than that of random forest and Bernoulli naïve Bayes, two-pass approach performs better in terms of F1 score compared to one-pass approach according to Figure 2.12. When F1 score of SVM is lower than the F1 scores of other models, one-pass and two-pass approaches showed no difference as shown Figure 2.14.

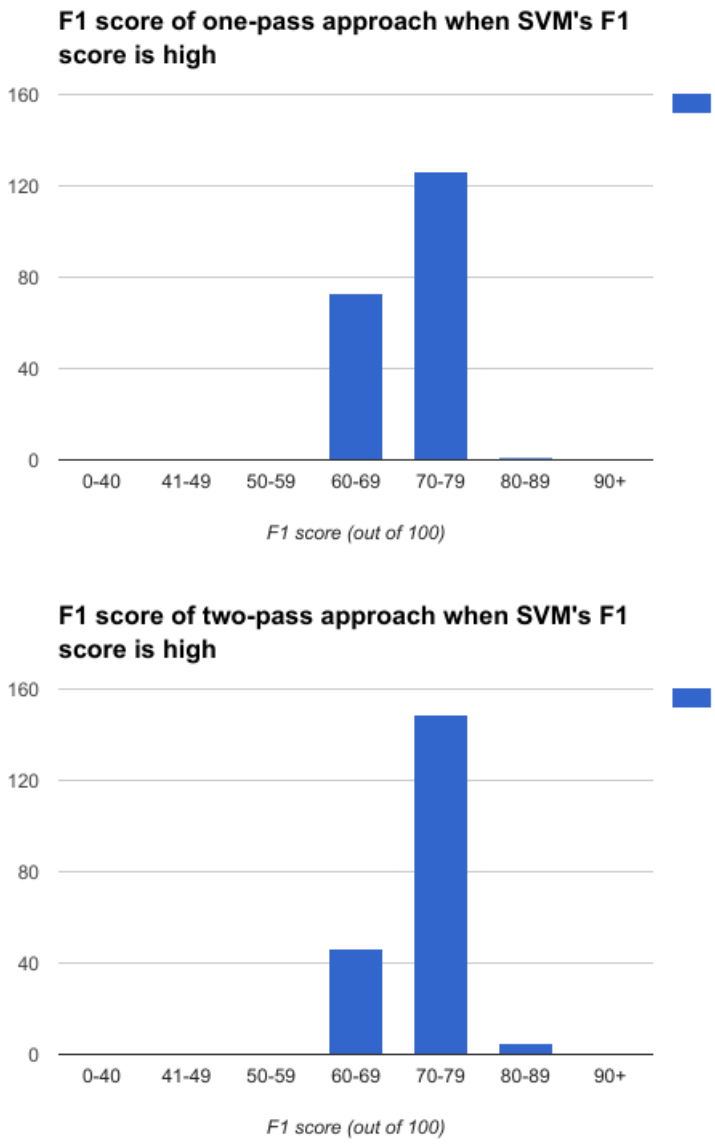


Figure 2.12 When SVM's F1 score is high, two-pass approach performs better

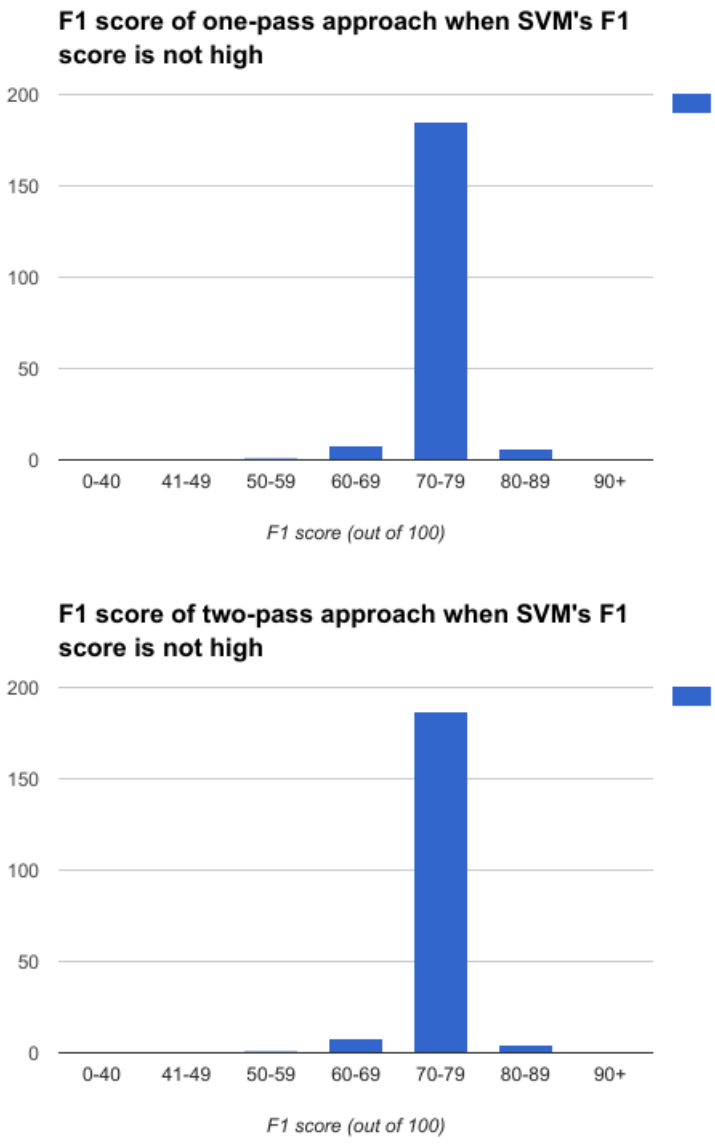


Figure 2.13 When SVM's F1 score is not high, both approaches performs on par

Table 2.13 shows the thresholds calculated from F1 score used to determine whether a prediction would be positive or negative after combining results from different models using formulas from (2.1.5 Step 5: Combined Model Prediction Generation).

Table 2.13 Two-pass approach in calculating prediction by combining predictions by the models

	Value	*Remark*
Threshold (first pass)	0.7010	If the sum of weight*prediction is larger than

		threshold, the result is taken as positive
Prediction (first pass)	1	$0.7353\times1 + 0.6667\times0 > 0.7010$ (Random forest and Bernoulli naïve Bayes')
Threshold (second pass)	0.7201	If the sum of weight*prediction is larger than threshold, the result is taken as positive
Prediction (final)	1	$0.7010\times1 + 0.7391\times0 > 0.7201$ (1st pass threshold and F1 scores of SVM)

Although many of the predictions were calculated using both first-pass and second-pass results, for some instances in test data that do not have a corresponding expert knowledge model result, the first-pass prediction was taken as the final prediction result instead. This behavior is caused by the inability to retrieval company LinkedIn profile in (2.2.1 Developing Company Information Crawler and Scraper) such that there was no corresponding expert knowledge vector in the expert knowledge data set.

It was found out that with Word2vec, the precision, recall and F1 score of Bernoulli naïve Bayes could be less than 0.5. In that case, the model is performing worse than random guessing which has precision, recall and F1 score 0.5. To avoid faulty model from greatly affecting the combined output, the weight of a model that has F1 score less than 0.5 would be set to 0.5, so that it is performing on par with random guessing instead of worse.

2.3 Testing

2.3.1 Testing Web Crawler and Scraper

Web crawlers and scrapers developed in this thesis are tested by their effectiveness in retrieving URLs and scraping content from these URLs.

2.3.1.1 Testing Company Website and Social Profile URL Retriever

To ensure the company website and social profile URL retrievers work, the outputs of the retrievers were examined.

For company website URL retriever, the output CSV has the following schema:

Website URL	Company name	Confidence	Valid

The website and company names were examined manually to see whether the websites match with the company names. A column called confidence was introduced and populated with a confidence value of the match. The confidence value was calculated by summing up the Levenshtein ratios of the combination of company name, URL and link text. An exact match would yield a confidence value of 3 in total. Generally, website URL with confidence value less than 1 are most possible to be invalid URLs. The invalid URLs were removed after manual examination.

For company social profile URL retriever, the output CSV has the following schema:

Website URL	Company	Social Network

The value of social network column is either of Facebook, LinkedIn or Crunchbase. For each company, there are three rows of records, each with an empty website URL value or a URL corresponding to the social profile URL of the respective network. Since a regular expression check have been done on the URLs after company name match, the social profile URL are guaranteed to be of that social network despite it may not always be the correct profile page of that company. Manual examination was performed to ensure the records are valid.

2.3.1.2 Testing Company Information and LinkedIn Scraper

To ensure the company information and LinkedIn scraper work, the outputs of the scrapers were examined.

For company information scraper, the data are difficult validate because descriptions of different companies do not exhibit a fixed pattern. The scraper may be misled by mal-formed website structure. For instance, the scraper may take all pages with an URL component '`about-us/`' as valid company description, despite there were many irrelevant pages under '`about-us/`'. Therefore, manual examination was performed to ensure the company information scraper works as expected.

For LinkedIn scraper, the data are guaranteed to be well-formed because of the fixed page layout of a LinkedIn company profile page. Although some profile pages may lack the website URL, industry, company size, or headquarter location fields, those are set to default values for expert knowledge qualification. Company description were generally successfully extracted from the LinkedIn page. Records with empty company descriptions had been automatically removed before the results were dumped onto an CSV file.

2.3.2 Testing Text Preprocessor

To test the text preprocessor, the output CSV file, was examined.

After text preprocessing, the output CSV has the following schema:

Description	Company

The description contained no terms listed in the stop word list and devoid of non-English terms. Terms with different suffixes had their suffixes stripped, leaving terms like ‘promot’ (promote), ‘colleagu’ (colleagues) and ‘consum’ (consume).

After examination of the CSV file, the text preprocessor was deemed to be able to output correct data.

2.3.3 Testing Classifiers

To test the predictions generated by the model, confusion matrix is used to assess accuracy of specific discrete classifier. Receiver operating characteristics (ROC) graph analysis [25] is also used to compare different classifiers.

2.3.3.1 Testing Classifier with Confusion Matrix

Given a binary classifier outputting a label in $\{Y, N\}$ as a prediction on given input. A prediction is considered *true positive* if the prediction matches the actual label. In a confusion matrix, cells corresponding to *true positives*, *false positives*, *false negatives* and *true negatives* are shown and using these four fields such that it is possible to calculate the rate precision, recall and accuracy of the model.

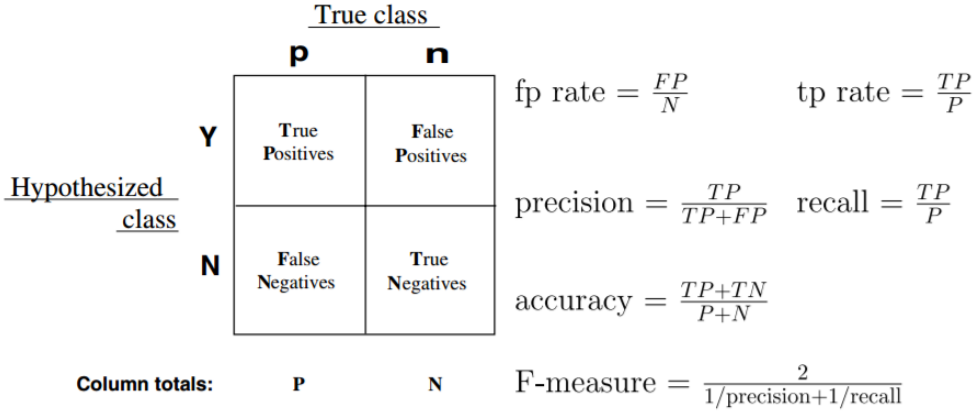


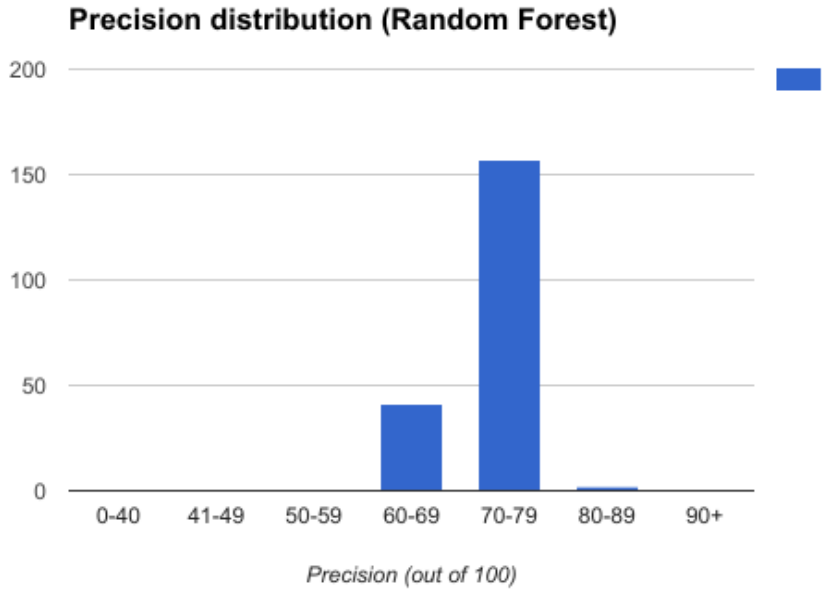
Figure 2.14. Confusion matrix and common performance metrics [25]

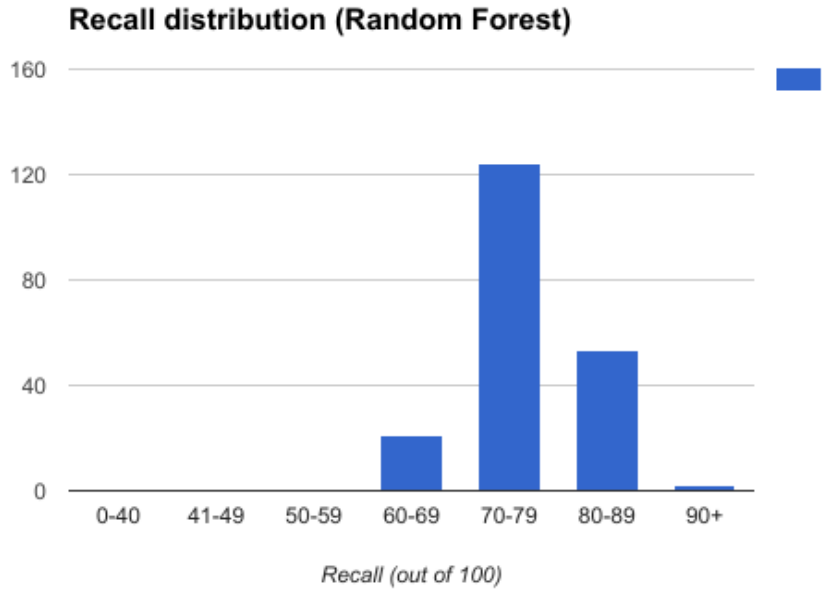
The confusion matrix is useful for testing classifier with only class decision such as SVM. For classifier outputting a continues value in form of probability ROC curve could also be used instead. For example, ensemble learning algorithm like random forest generates class probability internally and by setting different thresholds, a ROC curve can be drawn. They can be found in (2.3.4 Testing Classifier with ROC Curve).

However, the confusion matrix along is not enough for testing whether the model is effective or not. To test the classifiers using confusion matrix, 200 trainings on the same data set have been performed on the random forest, Bernoulli naïve Bayes and SVM models. Precision, recall and F1 score values have been calculated and recorded for each training respectively and their distributions have been drawn onto histograms in the follow sections.

2.3.3.1.1 Random Forest

Feature extraction and training-test data split involve randomness. The distribution histograms shown below represents the results of an instance of LSA training and training-test data split. The classifier achieved 0.7-0.79 precision, recall and F1 score out of 1.


Precision distribution (Random Forest)
200
150
100
50
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Precision (out of 100)


Recall distribution (Random Forest)
160
120
80
40
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Recall (out of 100)

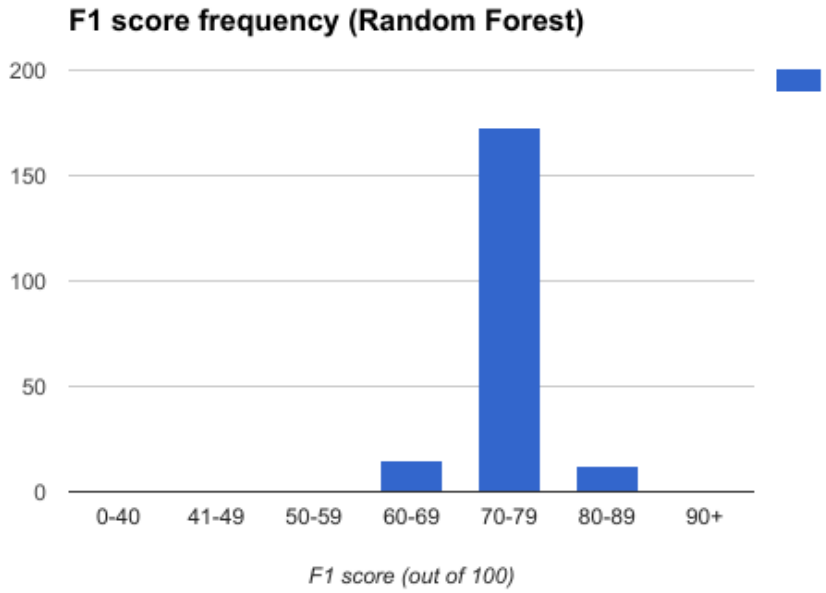
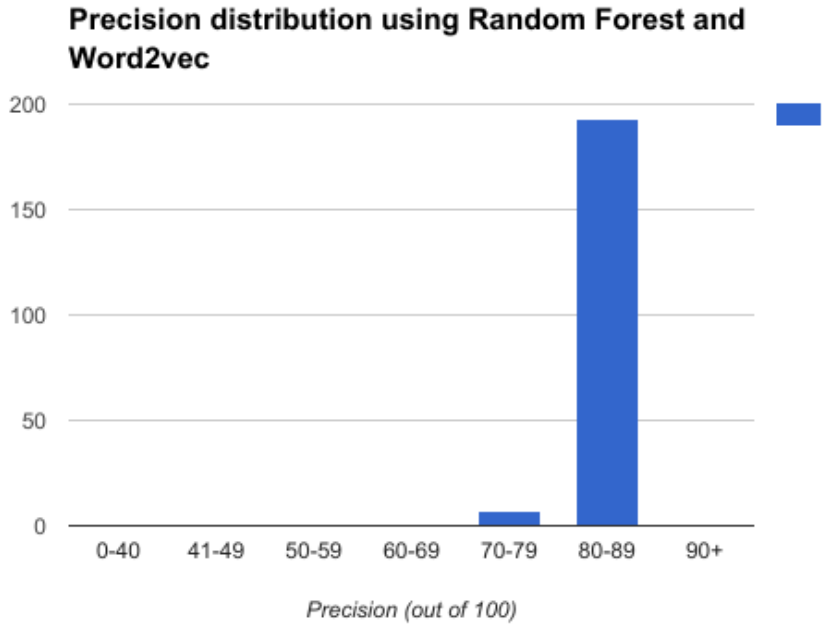
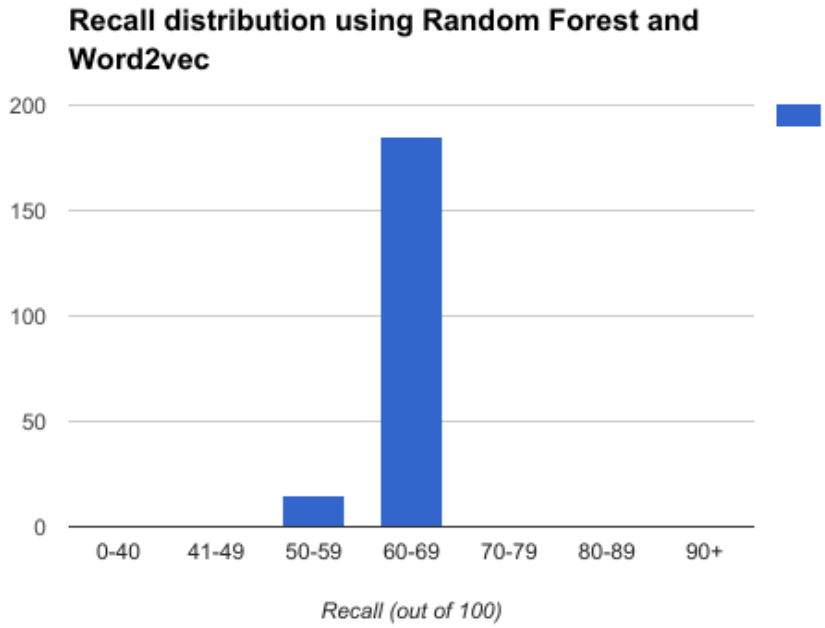


Figure 2.15 Precision, recall and F1 score of one instance of random forest model using LSA

The following histograms display an instance of Word2vec training. Word2vec was performing on par with random forest model trained on LSA-generated feature vectors having achieved 0.8-0.89 median precision, 0.6-0.69 recall and 0.7-0.79 median F1 score out of 1.


Precision distribution using Random Forest and Word2vec
200
150
100
50
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Precision (out of 100)


Recall distribution using Random Forest and Word2vec
200
150
100
50
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Recall (out of 100)

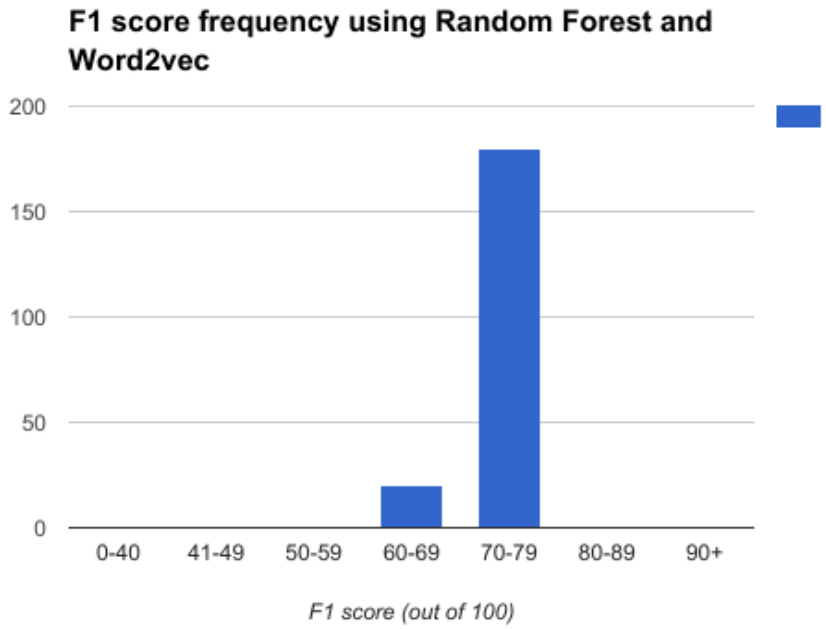


Figure 2.16 Precision, recall and F1 score of one instance of random forest model using Word2vec

2.3.3.1.2 Bernoulli naïve Bayes

Precision, recall and F1 score of a Bernoulli naïve Bayes model remain constant over 200 training because its accuracy depends only on the randomness of training-test data split and the LSA/Word2vec model. Therefore, there is no need to draw the histograms for Bernoulli naïve Bayes.

Using feature vectors generated from LSA with 20 output components, Bernoulli naïve Bayes performed on par with random forest, achieving 0.6857 precision, 0.8 recall and 0.7385 F1 score.

Table 2.14 Precision, recall, F1 score of one instance of Bernoulli naïve Bayes model using LSA

Precision	0.6857
Recall	0.8
F1 score	0.7385

Bernoulli naïve Bayes performs on par with random forest on LSA-generated feature vectors. However, when being trained using feature vectors generated by Word2vec, its F1 score decreased dramatically. The classifier achieved precision of 1.0, recall of 0.0976 and F1 score 0.1778. This is even worse than random guessing. When combining the final output shown in (2.3.3.1.4 Ensemble of Models), the weight of this rouge Bernoulli naïve Bayes model is set to 0.5 to emulate random guessing.

Therefore, it is not recommended to use Bernoulli naïve Bayes when the feature vectors are generated using Word2vec. Later discussion in (3.3 Effect of Feature Extraction Algorithm on Classifier Performance) discovered that K-nearest neighbors (K-NN) can generate much more accurate results and it should be used instead.

Table 2.15 Precision, recall, F1 score of one instance of Bernoulli naïve Bayes model using Word2vec

Precision	1.0
Recall	0.0976
F1 score	0.1778

2.3.3.1.3 SVM

Precision, recall and F1 score of a SVM model remain constant over 200 training just like Bernoulli naïve Bayes. Table 2.16 and Table 2.17 shows the SVM precision,

recall and F1 score that was combined with other models in (2.3.3.1.4 Ensemble of Models).

Table 2.16 Precision, recall, F1 score of one instance of SVM in training case using LSA

Precision	0.76
Recall	0.7308
F1 score	0.7451

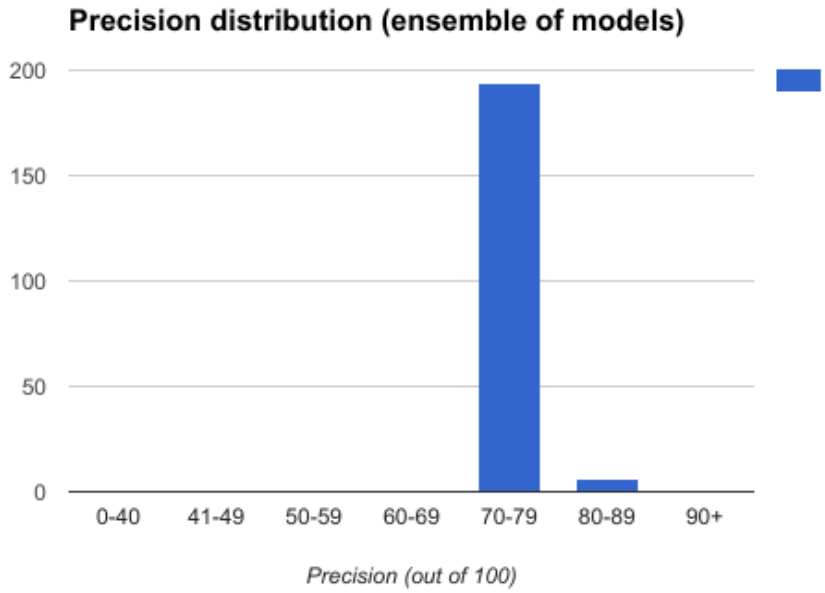
Table 2.17 Precision, recall, F1 score of one instance of SVM in training case using Word2vec

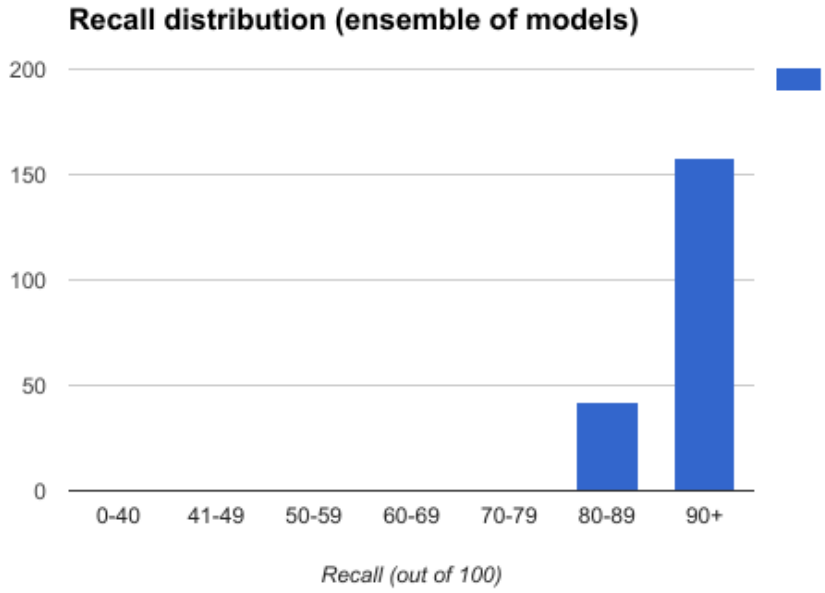
Precision	0.9167
Recall	0.6875
F1 score	0.7857

2.3.3.1.4 Ensemble of Models

Random forest, Bernoulli naïve Bayes and SVM using feature vectors generated from LSA and Word2vec are combined as described in (2.2.5 Generating Predictions from Models Combined). The precision, recall and F1 score of the outcome is better than any individual model because the models collectively decide which prediction is the most suitable. Even if one model predicts it wrong, the other two models can correct the error.

Using LSA, the combined output has precision median of 0.7-0.79, recall median of 0.9+ and F1 score median of 0.8-0.89 as shown in Figure 2.17.


Precision distribution (ensemble of models)
200
150
100
50
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Precision (out of 100)


Recall distribution (ensemble of models)
200
150
100
50
0
0-40
41-49
50-59
60-69
70-79
80-89
90+
Recall (out of 100)

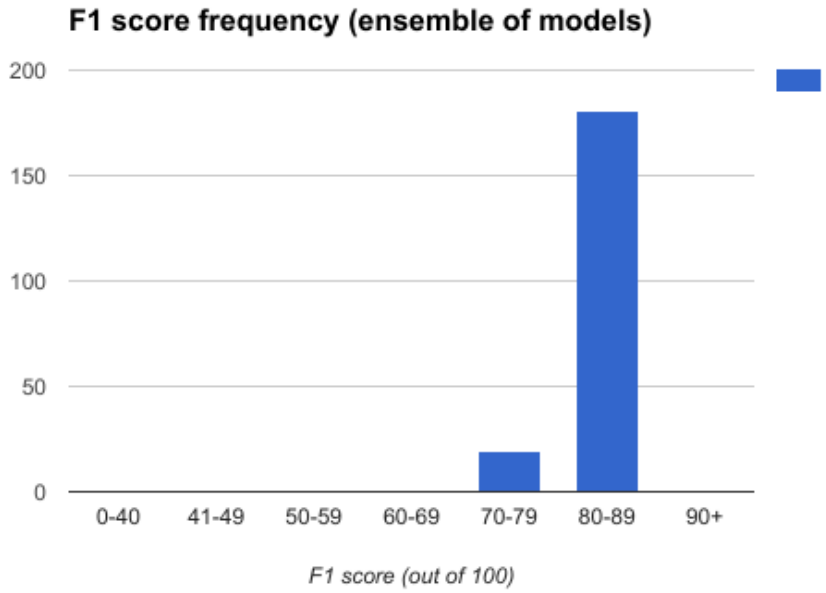
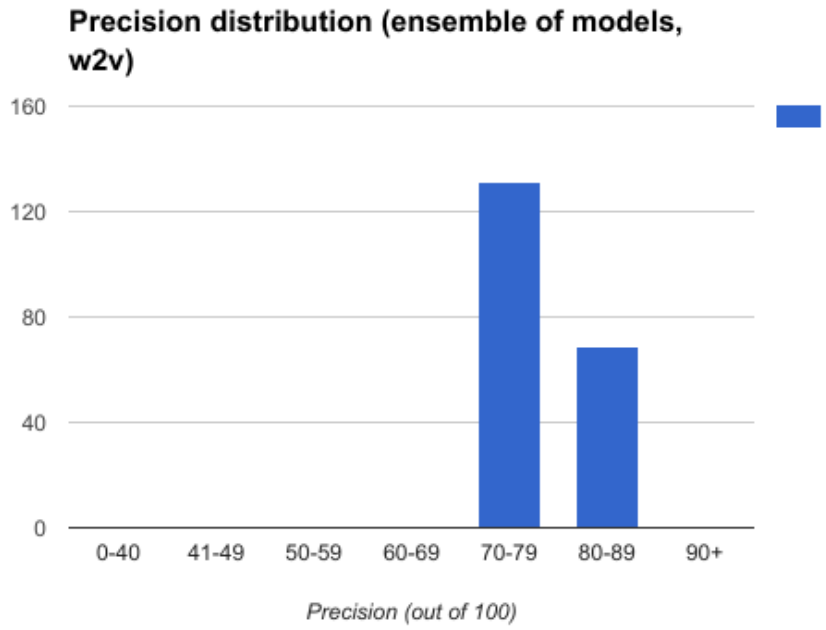


Figure 2.17 Precision, recall and F1 score distribution of combined models trained on LSA

Using Word2vec, the combined output has precision median of 0.7-0.79, recall of 0.6-0.69 and F1 score medial of 0.7-0.79 as shown in Figure 2.18.



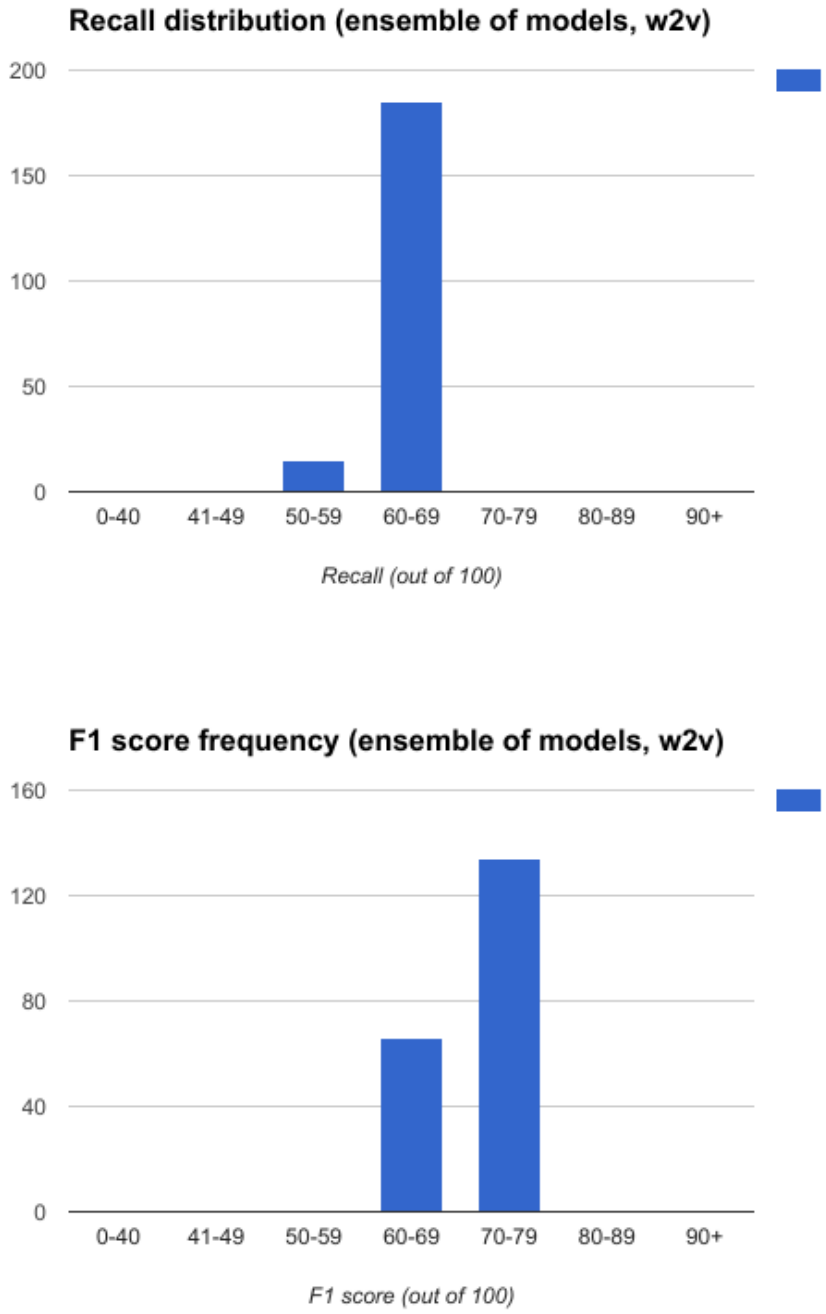


Figure 2.18 Precision, recall and F1 score distribution of combined models trained on Word2vec

2.3.4 Testing Classifier with ROC Curve

A ROC graph is a plot of *true positive rate* against the *false positive rate* given different class probability thresholds.

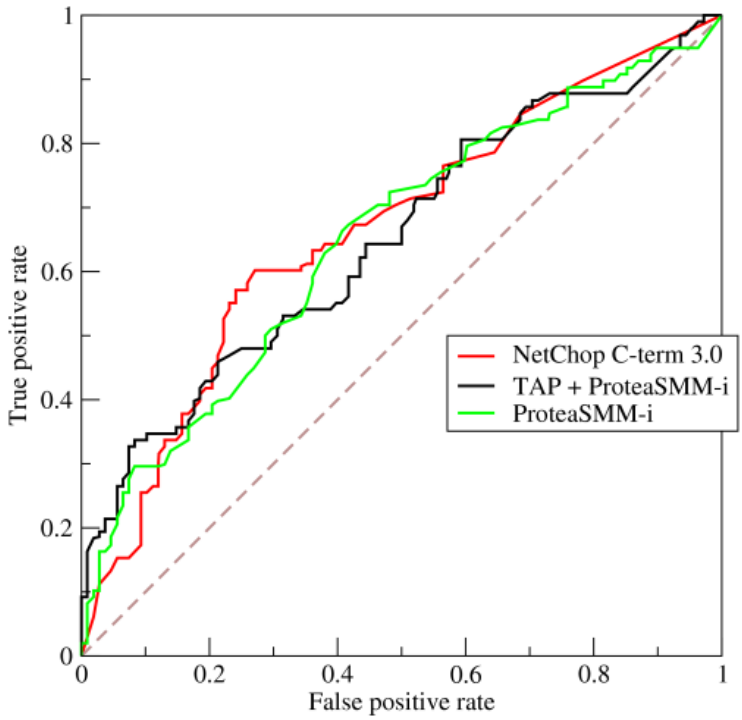


Figure 2.19. A ROC curve of three classifiers [31]

In Figure 4, the dotted line denotes a random guessing classifier. A random guessing classifier always has the same true positive and false positive rate. If the ROC curve of a classifier is below the dotted line, that means it performs worse than random guessing, which may mean the information given are falsely interpreted by the classifier. In contrary, the closer an ROC curve is closer to the top and left side, the more accurate it is.

ROC curves of different classifiers in this study are plotted and compared. In making conclusion of whether on classifier out-performs another, care should be taken as the ROC curves may differ given different test data sets. [25] suggests dividing data sets into *N* sets and plot the *N* ROC curves to calculate variance before comparing different classifiers.

The predictor developed in this study is expected to generate a ROC curve

comparable if not better than previous works [4], [5] and [6].

Ten ROC curves of Random forest classifier trained on LSA and Word2vec feature vectors are displayed in Figure 2.20 and Figure 2.21. The curves were made semi-transparent to allow for observing overlapping curves and each model has ten curves drawn on the same figure. The slanted dashed line on both figures represent the ROC curve of a random guessing classifier with AUC of 0.5.

At first glance it was found that when the ROC curve of Bernoulli naïve Bayes trained using Word2vec-generated feature vector is reassembling the dashed line, meaning its accuracy is close to a random guessing classifier. This echoes the inferior F1 score of Bernoulli naïve Bayes in (2.3.3.1.2 Bernoulli naïve Bayes).

The ROC curves of random forest models in both cases and the Bernoulli naïve Bayes trained using LSA-generated feature vectors are performing superior to the random guessing curve, meaning that they can generate meaningful predictions.

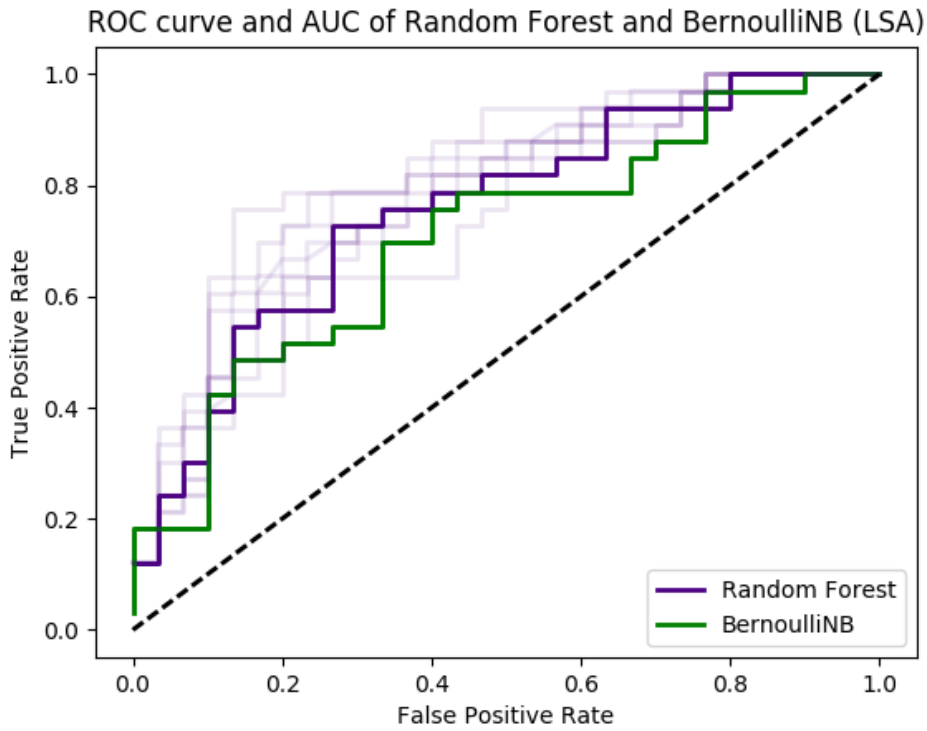


Figure 2.20 ROC curve of random forest and Bernoulli naïve Bayes classifier trained on LSA-generated feature vectors

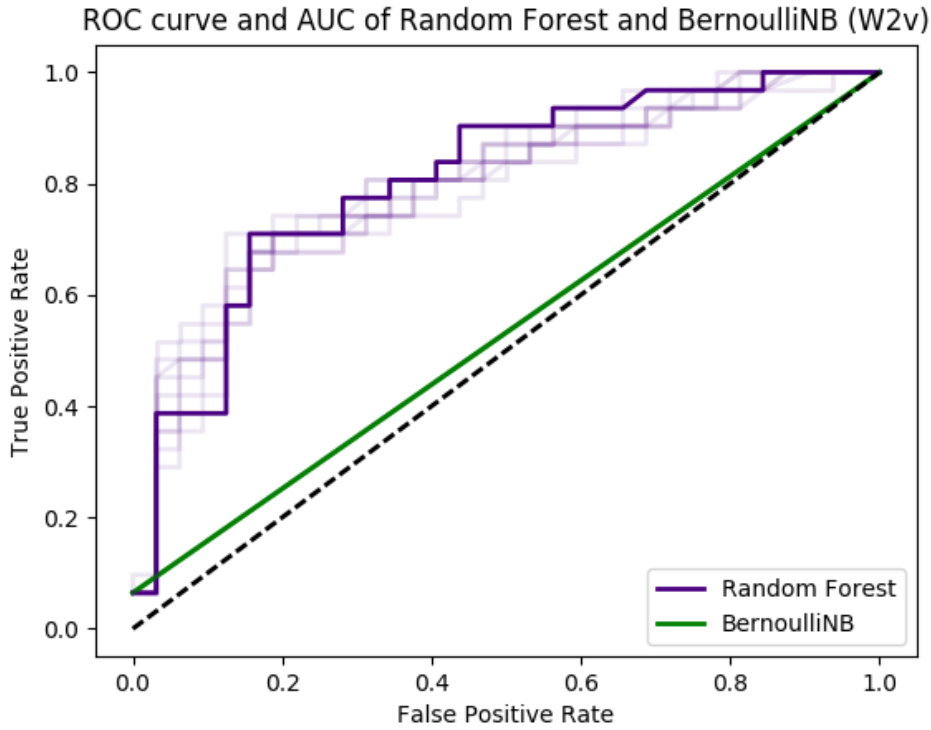


Figure 2.21 ROC curve of random forest and Bernoulli naïve Bayes classifier trained on Word2vec-generated feature vectors

Although SVM is not a probabilistic model that does not have probability distribution over the classes as it generates predictions, Platt scaling can be used to create per-class probability distribution for an SVM classifier [24]. This is done by fitting a logistic regression model with the logistic function:

$$P(y = 1|f(x)) = \frac{1}{1 + exp(Af(x) + B)} \tag{2.19}$$

where $f(x)$ is the classifier score $\in \{-1,1\}$ from a given sample x from the SVM and A, B are two scalar parameters learned by the logistics regression.

Using Platt scaling, the estimated ROC for the SVM classifier trained using expert knowledge data is displayed in Figure 2.22.

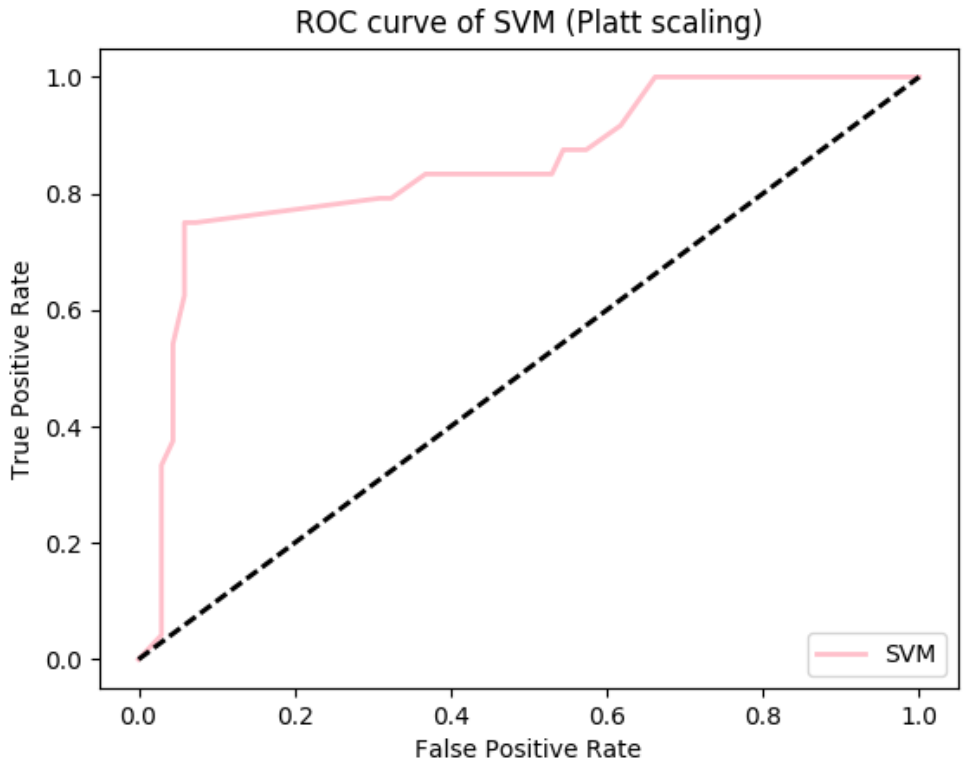


Figure 2.22 ROC curve of the SVM classifier

2.3.5 Testing Classifier with Area Under an ROC Curve (AUC)

Area under an ROC curve, as suggested in [7], can be calculated using trapezoidal integration:

$$\text{AUC} = \sum_i \left\{(1 - \beta_i \cdot \Delta\alpha) + \frac{1}{2}[\Delta(1-\beta) \cdot \Delta\alpha]\right\} \tag{2.20}$$

where

$$P(\text{false positive}) = \alpha \tag{2.21}$$

$$P(\text{true positive}) = 1 - \beta \tag{2.22}$$

$$\Delta(1-\beta) = (1-\beta_i) - (1-\beta_{i-1}) \tag{2.23}$$

$$\Delta\alpha = \alpha_i - \alpha_{i-1} \tag{2.24}$$

The use of trapezoidal integration is justified by the fact that all the points on the ROC curve are connected by straight lines instead of smooth curves.

For a perfect classifier, AUC should equal to 1.0 while any realistic classifier should not have AUC < 0.5 as the AUC of a random guessing classifier is 0.5. It is expected that the predictor developed in this study would give a AUC > 0.6.

The AUCs shown in this section are calculated from the ROC curves in (2.3.4 Testing Classifier with ROC Curve).

The AUC distributions of random forest trained on LSA and Word2vec shown in Figure 2.23 showed that the median AUC for random forest is 0.70-0.79 and 0.80-0.89 for LSA and Word2vec respectively. These results echo the ROC curves in (2.3.4 Testing Classifier with ROC Curve) and observations made on F1 scores in (2.3.3.1.1 Random Forest) that they perform with similar accuracies.

Figure 2.24 revealed that for Bernoulli naïve Bayes trained on Word2vec its AUC is close to 0.5 at 0.5323. On the other hand, AUC of Bernoulli naïve Bayes trained on LSA has AUC of 0.7071, meaning that the model can generate meaningful predictions.

Lastly, the AUC of an SVM the ROC curve in Figure 2.22 is 0.8474.

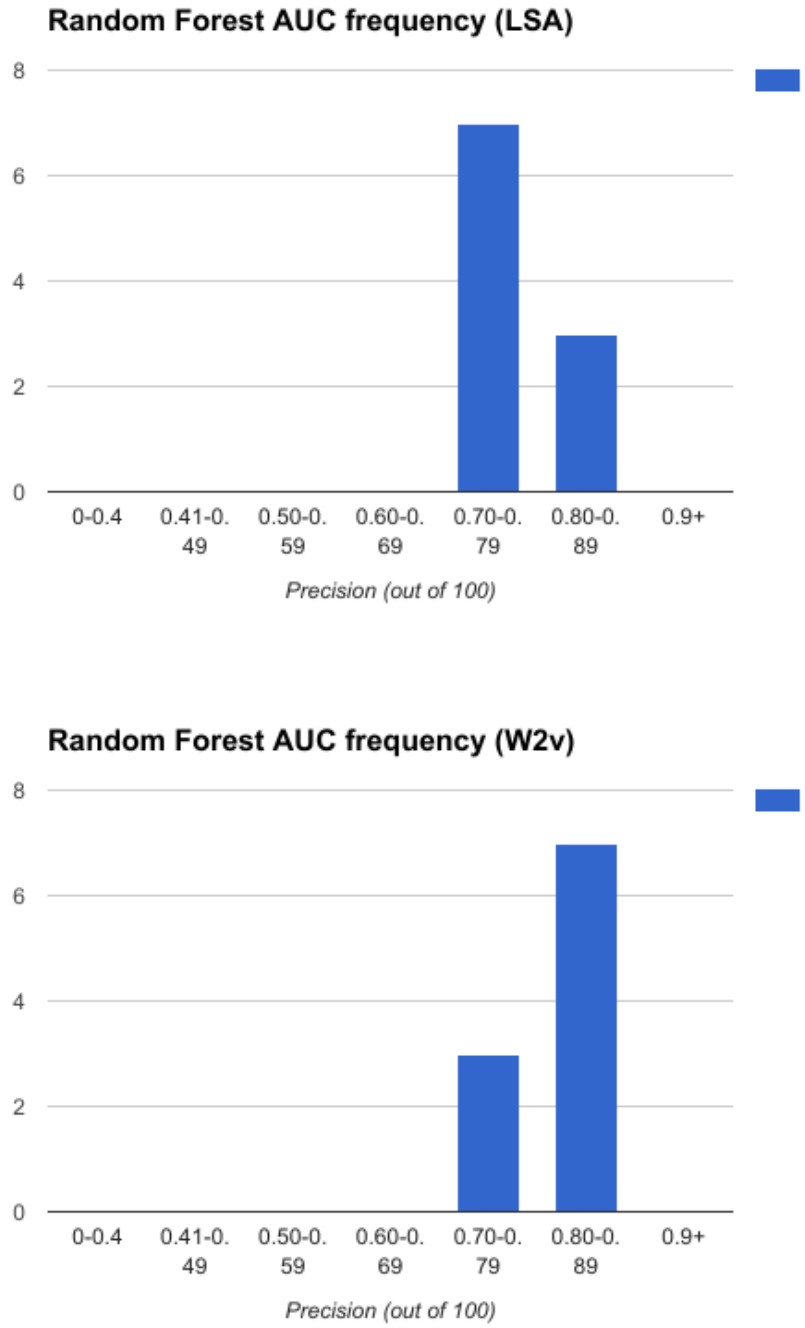


Figure 2.23 AUC distribution of random forest trained on LSA and Word2vec feature vectors

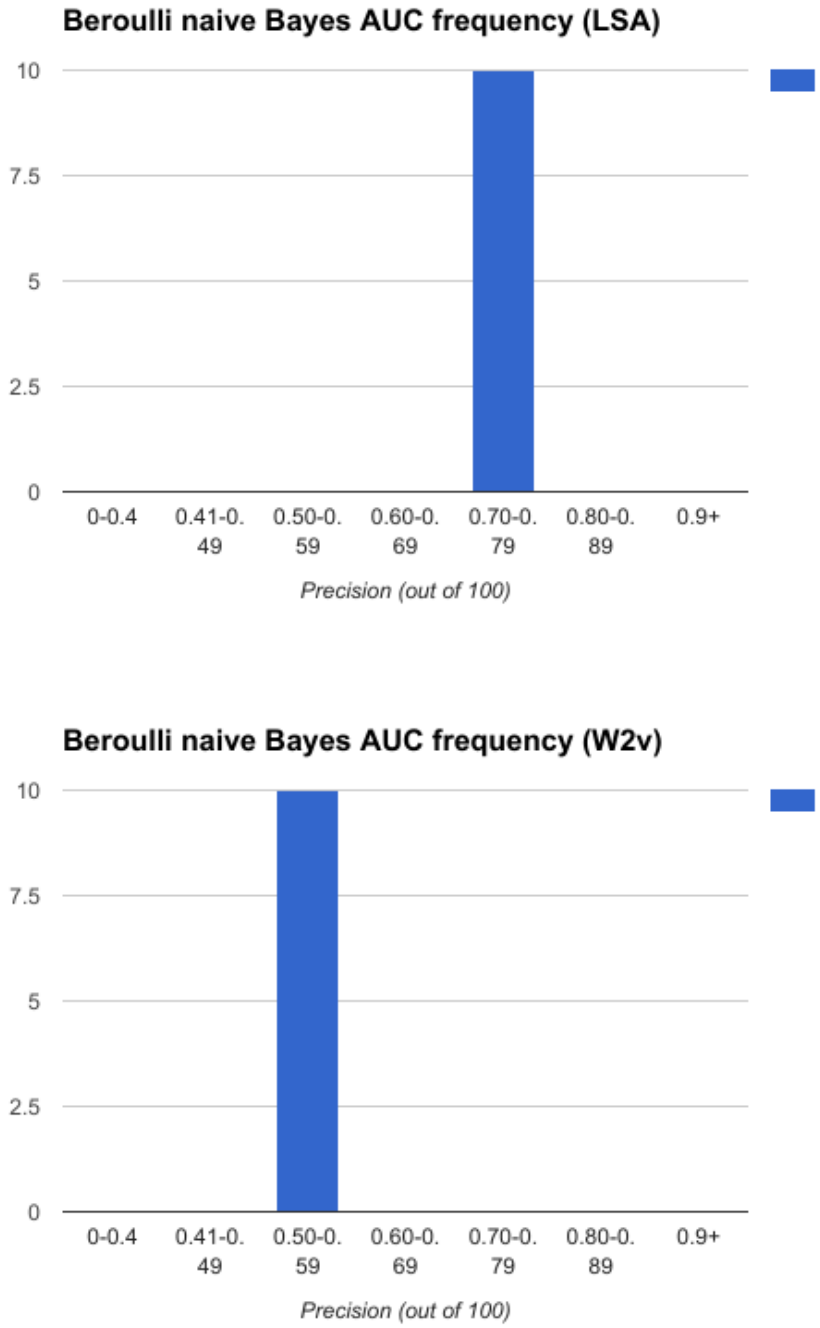


Figure 2.24 AUC distribution of Bernoulli naïve Bayes trained on LSA and Word2vec feature vectors

2.4 Evaluation

2.4.1 Company Website/Social Profile URL Retriever

To evaluate the company website URL retriever, the number of input and effective

output have been calculated. With the sales data obtained from EventXtra, the URL retriever retrieved 467 company website URLs and 1229 social network URLs from a list of 612 company names. Out of 467 company website URLs, 37 was considered invalid result after manual review. The URL retriever could retrieve 70% of the website URLs automatically. Therefore, it has successfully achieved the objective of automatically retrieving company website and social profile URLs. Additional details could be found in Table 2.18.

Table 2.18 Number of company website and social profile URLs retrieved

	Qualified	Disqualified	Total
Source Company Name	170	442	612
Website	136	331	467
Invalid Website	22	15	37
Facebook Profile	129	332	461
LinkedIn Profile	112	284	396
Crunchbase Profile	100	272	372

2.4.2 Company Website/Social Profile Scraper

To evaluate the company information and LinkedIn scraper, the number of input and effective output were calculated as in (2.3.1.1). Out of 467 company websites visited, 1072 lines of company descriptions were scraped including 721 lines from the company website itself and 351 lines from LinkedIn profile page. The scrapers have showed that they are able to automatically scrape company website and LinkedIn profile page for required information, so they have successfully achieved the objective as web scrapers. Details could be found in Table 2.19.

Table 2.19 Number of company website and social profile scraped

	Qualified	Disqualified	Total
Source Company Website	136	331	467
Source LinkedIn	112	284	396
Scraped Webpages	93	628	721
Scraped LinkedIn pages	92	259	351

2.4.3 Process Data and Prediction with Ensembles of Models

One objective of this thesis is to create a business lead qualifier that integrates legacy sales data and expert knowledge using data scraped online to predictively qualify leads. To quantify and evaluate the effectiveness of the lead qualified created, matrices namely precision, recall, F1 score and AUC are used. The precision, recall, F1 score and AUC of a random guessing classifier is also included in the tables below for reference.

Table 2.20 Precisions, recalls, F1 scores and AUCs of different classifiers under different feature extraction method

		Precision	Recall	F1 score	AUC
	Random Guessing	0.5	0.5	0.5	0.5
LSA	**Random forest**	0.7-0.79	0.7-0.79	0.7-0.79	0.7-0.79
	Bernoulli naïve Bayes	0.6857	0.8	0.7385	0.7-0.79
	SVM	0.76	0.7308	0.7451	0.8474
	Ensemble of models	0.7-0.79	0.9+	0.8-0.89	N/A
Word2vec	**Random forest**	0.8-0.89	0.6-0.69	0.7-0.79	0.8-0.89
	Bernoulli naïve Bayes	1.0	0.0976	0.1778	0.5-0.59
	SVM	0.9167	0.6875	0.7857	0.8285
	Ensemble of models	0.7-0.79	0.6-0.69	0.7-0.79	N/A

According to the data in Table 2.20, the precisions, recalls, F1 scores and AUCs of every classifier have performed much better than random guessing except for Bernoulli naïve Bayes trained on Word2gvec. This means the classifiers built for this thesis can generate meaningful predictions. Therefore, the objective of creating a business lead qualifier has been achieved.

2.4.4 Extend Previous Work by Incorporating Alternative Data Source and Algorithm

One objective of this thesis is to extend previous works by using different feature extraction method and by incorporating information from social network profile.

2.4.4.1 Using Alternative Feature Extract Method

In this thesis, Word2vec was chosen as an alternative to the commonly used LSA for

extracting feature vectors from textual data. Table 2.21 shows the comparison of different feature extraction methods and their respective median F1 scores when used to train different classifiers. Random Forest shows comparable results for both methods but Bernoulli naïve Bayes has exceptionally low F1 score when trained using Word2vec.

Table 2.21 Median F1 score of classifiers using different feature extraction methods

		Feature extraction method	
		LSA	Word2vec
Median F1 score	Random Forest	0.7-0.79	0.7-0.79
	Bernoulli naïve Bayes	0.7385	0.1778
	Ensemble of models	0.7-0.79	0.7-0.79

Table 2.22 AUCs of classifiers using different feature extraction methods

		Feature extraction method	
		LSA	Word2vec
AUC	Random Forest	0.8-0.89	0.7-0.79
	Bernoulli naïve Bayes	0.7-0.79	0.5-0.59
	SVM	0.8474	0.8285

The result of inferior F1 score when Bernoulli naïve Bayes classifier was trained on Word2vec-generated feature vectors was due to high precision and low recall according to Table 2.15. Bernoulli naïve Bayes algorithm missed most of the positive case but when it asserts a positive case it is very likely to be correct. This is further discussed in (3.3 Effect of Feature Extraction Algorithm on Classifier Performance).

Despite Bernoulli naïve Bayes underperforms when being trained with Word2vec-generated feature vectors, the eventual median F1 score of ensembles of models at 0.7-0.79, on par with models trained on LSA while all of them are performing better than a random guessing classifier with 0.5 F1 score. When comparing the individual AUC of random forest, Bernoulli naïve Bayes and SVM classifiers with the results in [4] and [6], they are inferior to the AUC (0.99985-1.0) in [6] while superior to the AUC (0.6116-0.6352) in [4]. Therefore, this thesis successfully created an effective business lead qualifier with alternative feature extraction algorithm Word2vec.

2.4.4.2 Using Alternative Data Source

Another aspect this thesis extends previous works is by incorporating information from social network profiles. While in [5], expert knowledge qualification was done exclusively by scraping the company websites, this thesis uses both company website and social network LinkedIn to qualify some expert knowledge qualification for the companies. Despite in [5] AUC was mainly used to evaluate the classifier and it is difficult to compare apple and orange, in this case to compare AUC with F1 score (SVM does not have an AUC), with a median F1 score in the 0.7-0.79 range, it can be concluded that the use of social network has improved the overall effectiveness of an expert knowledge classifier.

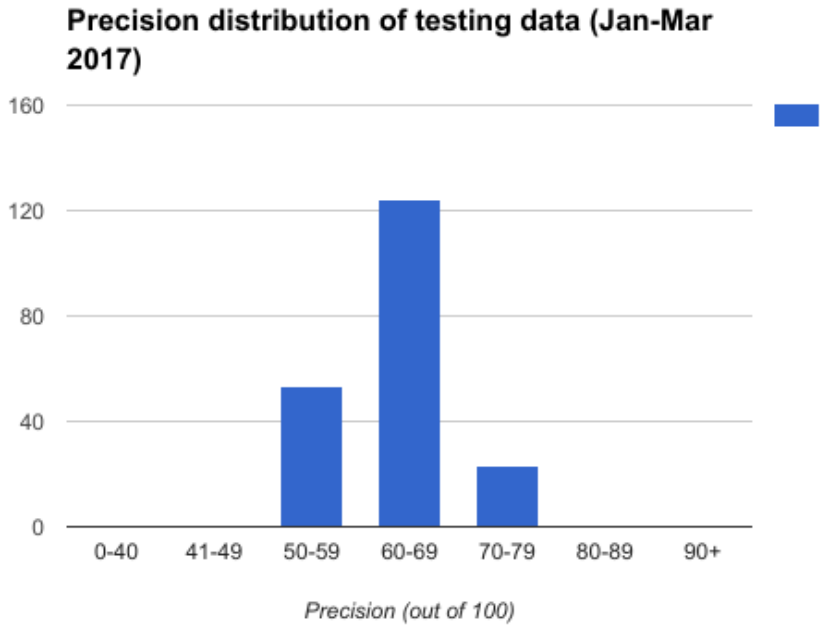
2.4.5 Practical Evaluation of the Lead Qualifier

A partnering Hong Kong company EventXtra Inc. agreed to assist the evaluation of the predictor developed in this study.

The models were originally trained on data from February 2014 to February 2017. To

assess its effectiveness on newer data, sales data was obtained from EventXtra. The test data set in this evaluation contains sales data from January 2017 to March 2017 because otherwise the testing data set would be too small.

200 training and testing had been performed on the testing data. Precision, recall and F1 score of the ensemble of models have been calculated and shown in Figure 2.25 Precision, recall and F1 score of ensemble of models over 200 trainings. The median for all three measures are 0.6-0.69 trained on LSA, which is slightly less than that from the previous data obtained from training set, which are 0.7-0.79.



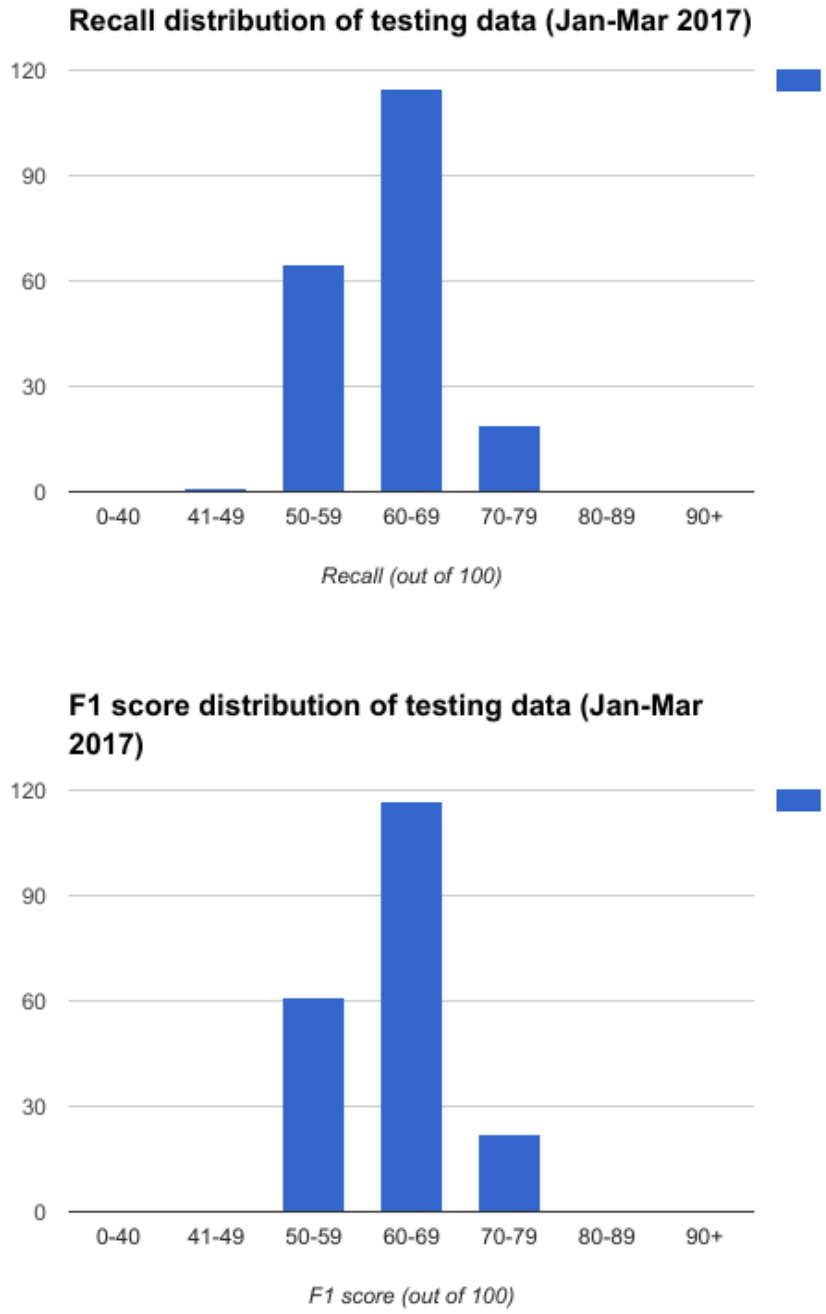
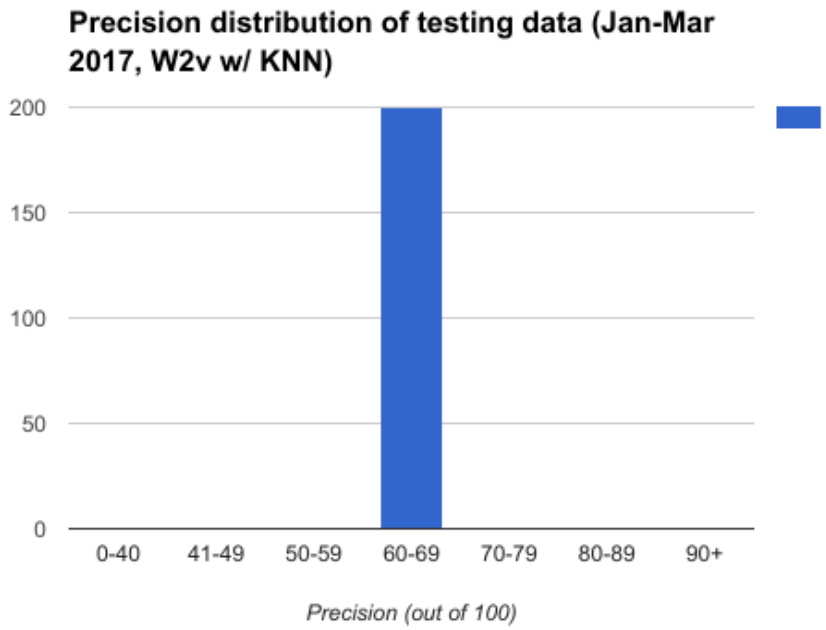
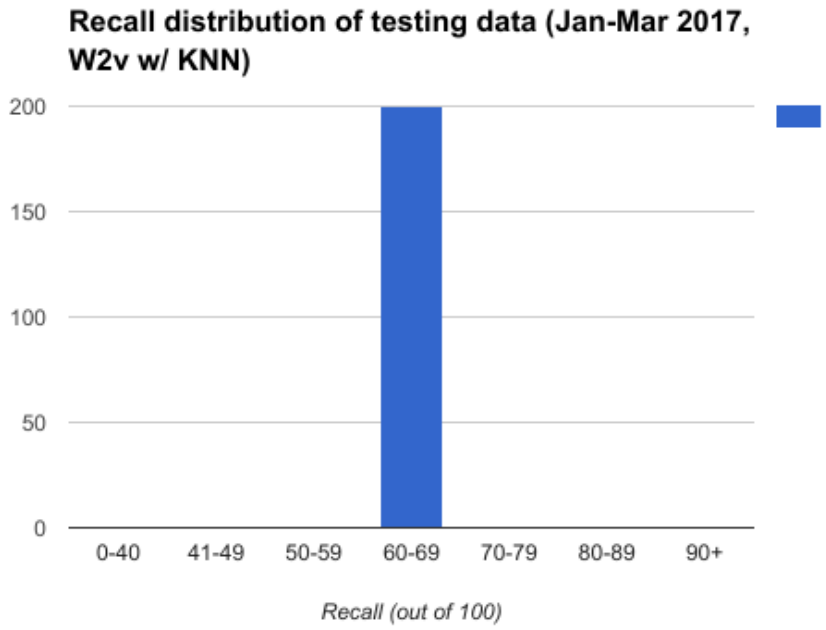


Figure 2.25 Precision, recall and F1 score of ensemble of models over 200 trainings

When trained using Word2vec, the precision, recall and F1 score of the Bernoulli naïve Bayes model dropped to 0.0 and the script automatically fell back to use random guessing classifier score (0.5) as weight for the Bernoulli naïve Bayes model.

However, this way the ensemble of model failed because one of the model fell back to a random guessing classifier. The performance improved after replacing Bernoulli naïve Bayes model with a K-nearest neighbors classifier according to Figure 2.26.





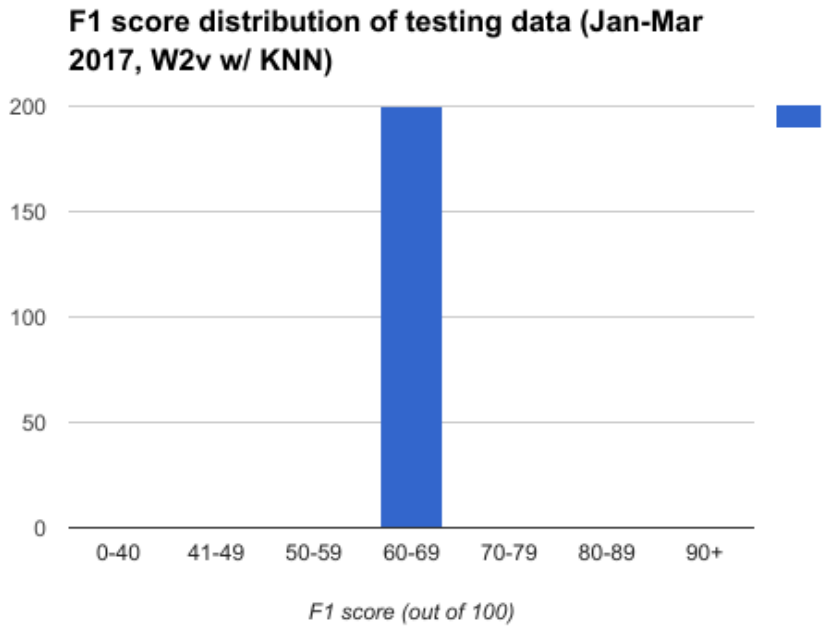


Figure 2.26 Precision, recall and F1 score of ensemble of models over 200 trainings, replacing Bernoulli naïve Bayes with K-nearest neighbors model

In conclusion, with two different ensembles of models, the classifier has been able to qualify business leads with meaningful outputs using existing sales data with median F1 score ranging from 0.6 to 0.79.

3 Discussion

3.1 Web Scraping on Search Engine and Sites

There were two major obstacles encountered during the development of the web crawlers and scrapers used in this thesis. The first one is related to getting blocked by DuckDuckGo due to high speed crawling. The second one is about LinkedIn page structure change on March 2017 that made previous version of the scraper useless.

At first, the web crawling script sending search request to DuckDuckGo had no throttling. After the first few attempts to crawl DuckDuckGo, the IP of the computer used to crawl DuckDuckGo was apparently blocked. Since DuckDuckGo serves an enormous number of users at any given time, high-speeding crawling serves as a kind of denial-of-service attack (DoS attack) that affect others from accessing the site because the site was busy responding to request from our side. This is an ethical issue on top of the existing technical issue. Throttling has since then been implemented to every crawler and scraper used in this thesis to avoid being blocked again and to protect the right of others to fairly use the search service by DuckDuckGo.

Secondly, there was a change in page layout and design in LinkedIn's information are hidden inside the multiple JSON-like string on the page instead of available in plain sight like the previous design. Python script had to be created to look for keywords in the page source code, then use those keywords to identify the correct HTML code to be extracted from the page. The scraper needs to be changed the next time when LinkedIn change their webpage structure for it to remain usable.

The research of this thesis was faced with the technical issues of rate-limiting and change in target website page structure on web crawling and scraping. I also encountered the ethical issue which requires the compromise of reducing crawling speed to better address the issue.

3.2 Corrective Effect of Ensemble of Models

In (2.3.3.1.4 Ensemble of Models), Bernoulli naïve Bayes was underperforming with F1 score of 0.1778, meaning it is more inaccurate than random guessing, which has constant F1 score of 0.5. However, the resultant F1 score of ensemble of models in the same instance reached a median of 0.7-0.79, as if there was no negative effect from the underperforming Bernoulli naïve Bayes.

This was because of the corrective effect of ensemble of models. The weight of how much a model contribute to the processing of combining predictions of many models is decided by the F1 score. According to Equation (2.13) and Equation (2.14), the Bernoulli naïve Bayes would have very little effect on the final prediction because other models are having weight much higher than the weight of Bernoulli naïve Bayes model, which was by then 0.5 as specified by Equation (2.13). Thanks to this corrective effect, the prediction of an ensemble of models is accurate even if one of the models has high error rate.

F1 score can be calculated for any binary classifier while AUC used in [6] cannot be used in non-probabilistic models like SVM except through Platt scaling. Therefore, the method of combining model predictions in this thesis is more general and more

readily applied to various kinds of classifier.

3.3 Effect of Feature Extraction Algorithm on Classifier Performance

In (2.3.3.1.2 Bernoulli naïve Bayes), Bernoulli naïve Bayes' performance was largely different given different feature extraction algorithms. The differences are summarized in Table 3.1.

Table 3.1 Comparison for Bernoulli naïve Bayes classifier performance on different feature extraction methods

	Precision	Recall	F1 score	AUC
LSA	0.6857	0.8	0.7385	0.7-0.79
Word2vec	1.0	0.0976	0.1778	0.5-0.59

Bernoulli naïve Bayes classifier underperformed on Word2vec-generated feature vectors because what Word2vec generates are word embeddings. Unlike LSA, which calculates term distribution according to their frequencies within the document, Word2vec generates word embeddings that are the relative positions of terms according the structures of the document instead of by their occurrence in the document like LSA did. Therefore, the vectors generated by Word2vec contains vectors clustering to when they have similar terms surrounding the term.

The algorithm of Bernoulli naïve Bayes focuses on the existence of terms inside of document instead of their relative position according to (2.1.4.1 Model Training with Random Forest and Naïve Bayes). Bernoulli naïve Bayes identified term clusters that

would qualify a business lead while missed many, which possibly were mixed cluster of qualified and disqualified terms, causing its high precision and low recall. Therefore, when given vectors representing relative positions between terms it performs inferiorly.

To verify this argument Bernoulli naïve Bayes was replaced by K-nearest neighbors, which classify the input geometrically by the known class of nearby neighbors. Table 3.2 shows that K-nearest neighbors classifier outperforms Bernoulli naïve Bayes with other factors unchanged, thus verifying that Bernoulli naïve Bayes is incompatible and K-nearest neighbors is compatible with Word2vec with the data set of this thesis.

Table 3.2 Precision, recall, F1 score of an K-nearest neighbors classifier using Word2vec

Precision	0.7000
Recall	0.7241
F1 score	0.7119

3.4 Two-pass Prediction

According to (2.2.3 Comparing One-pass and Two-pass Approaches), generating prediction from ensemble of models in two passes gave better median F1 score when the F1 score of the SVM was high. This can be explained by the corrective effect ensemble of models in a way by grouping classifiers of lower precisions to the first pass and classifiers with higher precisions in second pass. The corrective effect would correct some errors from classifiers of lower precisions as illustrated in the scenarios below using example data shown in Table 3.3.

Table 3.3 Example F1 scores and predictions by random forest, Bernoulli naïve Bayes and SVM

	Random forest	*Bernoulli naïve Bayes*	*SVM*
F1 score	0.7188	0.5806	0.7636
Actual Class	0 (negative)		
Scenario (a)	1 (positive)	1 (positive)	0 (negative)
Scenario (b)	1 (positive)	0 (negative)	0 (negative)

Table 3.4 illustrates how prediction is derived in one pass. That is, all F1 scores and predictions are aggregated using only one threshold calculated from the three F1 scores. The results were unsatisfactory because in both scenarios, the final predictions are all incorrect.

Table 3.4 Derivation of combined prediction result in one pass

	Value	*Remark*
Threshold (a)	0.6877	$\frac{0.7188 + 0.5806 + 0.7636}{3}$
Prediction (a) (incorrect)	1	$0.7188 + 0.5806 + 0.7636 \times 0 > 0.6877$
Threshold (b)	0.6877	$\frac{0.7188 + 0.5806 + 0.7636}{3}$
Prediction (b) (incorrect)	1	$0.7188 + 0.5806 \times 0 + 0.7636 \times 0 > 0.6877$

Table 3.5 illustrates how prediction is derived in two pass. This time, the results are correct for both scenarios. In scenario (a), the second pass even demonstrated how the error created in the first pass can be corrected in the second pass.

Table 3.5 Derivation of combined prediction result in two pass

	Value	*Remark*
Threshold (a) (1st pass)	0.6497	$\frac{0.7188 + 0.5806}{2}$
Prediction (a) (1st pass)	1	$0.7188 + 0.5806 \times 0 > 0.6497$
Threshold (a) (2nd pass)	0.7067	$\frac{0.6497 + 0.7636}{2}$
Prediction (a) (correct)	0	$0.6497 + 0.7636 \times 0 < 0.7067$
Threshold (b) (1st pass)	0.6497	$\frac{0.7188 + 0.5806}{2}$
Prediction (b) (1st pass)	0	$0.7188 + 0.5806 \times 0 > 0.6497$
Threshold (b) (2nd pass)	0.7067	$\frac{0.6497 + 0.7636}{2}$
Prediction (b) (correct)	0	$0.6497 \times 0 + 0.7636 \times 0 < 0.7067$

Compared to random forest and Bernoulli naïve Bayes, the input vectors of an SVM was less noisy so that the F1 score of an SVM is generally high. In a one-pass approach, the actual weight that SVM has on deciding the final prediction became less and the corrective effect was thus less apparent while two-pass approach, SVM is able to correct errors made in the first pass. Therefore, the two-pass approach is more fault tolerant and generates better result in general.

3.5 Effect of Imbalanced Training Sets

Originally, 104 qualified and 625 disqualified records were used to train the models. The precisions, recalls and F1 scores of each classifier were recorded from this training data set:

Table 3.6 Precisions, recalls and F1 scores of classifiers being trained on 104 qualified and 625 disqualified data

	Random Forest	Bernoulli naïve Bayes
Precision	1.0	1.0
Recall	0.0811	0.1351
F1 score	0.1500	0.2381

The precision was high while recall very low, meaning that the model misses out large part of the qualified data. The imbalance in data set (1 qualified:6 disqualified) caused many qualified item to be overlooked because of excessive noise from disqualified data. This situation was resolved by keeping the number of qualified and disqualified training data in balance of around 100 each.

Table 3.7 Precisions, recalls and F1 scores of classifiers being trained on 104 qualified and 105 qualified data

	Random Forest	Bernoulli naïve Bayes
Precision	0.8	0.6857
Recall	0.7059	0.7058
F1 score	0.75	0.6957

After balancing the number of qualified and disqualified data in the training data set, the precisions, recalls and F1 scores of both classifier improved dramatically.

4 Conclusion

4.1 Summary of Work

In this thesis, I used real sales data from a company and created automated lead qualifier with the data set by scraping online information about the companies listed in the data and process them into formats that are suitable for machine learning. The company website URL retriever successfully achieved 70% accuracy. Then, I combined multiple machine learning algorithms including random forest, Bernoulli naïve Bayes and SVM while discarding Gaussian naïve Bayes due to its relative suitability to the nature of the data and the actual performance. This approach has shown an overall improvement over single-model approach as shown in (2.4.3 Process Data and Prediction with Ensembles of Models).

This way, the lead qualifier scored a satisfying result of median F1 score of 0.7-0.79 and median AUC as high as 0.80-0.89 out of the 1.0 scale. These numbers are exciting because they are much higher than a random guessing classifier (constant 0.5 F1 score and 0.5 AUC).

4.2 Future Development

At the end of this work, there are two aspects of improvements that can be tackled in future researches.

4.2.1 More Reliable Expert Knowledge Qualification

Although most of the expert knowledge qualifications have been based on reliable data source like LinkedIn for company size, location and industry category, some of

which like the existence of contact information or whether the company organizes event are qualified though the existence of specific keywords in the company website. Expert knowledge qualification could be performed more intelligently by actively searching on search engines and social network, as a human would.

Expert knowledge has improved the efficiency of the qualifier by around 0.1+ in F1 score, yet there is no hard evidence of which combination of the expert knowledge would yield the best classification result because expert knowledge is simply the gut feeling from sales on their experience. More work could be done on automatically discovering the link between some attributes and the result of qualification via deep learning.

4.2.2 Multilingual Support

Many of the company concerned in the data set are having Chinese name or have Chinese-only company descriptions such that they are ignored in the early stage of the web scraping process because this thesis requires English company descriptions and most of the programming are optimized for English sites. For example, when comparing company name and the URL during company website retrieval, Chinese company name would cause many correct URLs to be discarded. With enough data, multiple models could be devised, each for a specific language, making this a more universal lead qualifier while not wasting part of the training data with non-English websites.

5 References

[1] fileboard, "Traditional Lead Scoring vs Predictive Lead Scoring," 10 February 2017. [Online]. Available: https://www.fileboard.com/traditional-lead-scoring-vs-predictive-lead-scoring/. [Accessed 10 February 2017].

[2] M. M. Long, T. Tellefsen and J. D. Lichtenthal, "Internet integration into the industrial selling process: A step-by-step approach," *Industrial Marketing Management,* vol. 36, no. 5, pp. 676-689, 2006.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems,* pp. 3111-3119, 2013.

[4] D. Thorleuchter, D. Van den Poel and A. Prinzie, "Analyzing existing customers' websites to improve the customer acquisition process as well as the profitability prediction in B-to-B marketing," *Expert Systems with Applications,* vol. 39, no. 3, pp. 2597-2605, 2012.

[5] J. D'Haen, D. Van den Poel, D. Thorleuchter and D. Benoit, "Integrating expert knowledge and multilingual web crawling data in a lead qualification system," *Decision Support Systems,* vol. 82, pp. 69-78, 2016.

[6] J. D'Haen and D. Van den Poel, "Model-supported business-to-business prospect prediction based on an iterative customer acquisition framework," *Industrial Marketing Management,* vol. 42, no. 4, pp. 544-551, 2013.

[7] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of

machine learning algorithms," *Pattern Recognition,* vol. 30, no. 7, pp. 1145-1159, 1997.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science,* pp. 391-407, 1990.

[9] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv,* 2013.

[10] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics dokladyLevenshtein, Vladimir I.,* vol. 10, no. 8, p. Soviet physics doklady, 1966.

[11] FullContact, "Company API - FullContact," [Online]. Available: https://www.fullcontact.com/developer/company-api/. [Accessed 12 February 2017].

[12] R. Boulton, "An English stop word list," [Online]. Available: http://snowball.tartarus.org/algorithms/english/stop.txt. [Accessed 15 2 2017].

[13] "Snowball," [Online]. Available: http://snowballstem.org/. [Accessed 12 February 2017].

[14] M. Porter, "An algorithm for suffix stripping," *Program,* pp. 130-137, 1980.

[15] G. Salton, A. Wong and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM,* vol. 18, no. 11, pp. 613-620, 1975.

[16] Y.-H. Chen, E. J.-L. Lu and M. F. Tsai, "Finding keywords in blogs: Efficient keyword extraction in blog mining via user behaviors," *Expert Systems with Applications,* vol. 41, no. 2, pp. 663-670, 2014.

[17] P. W. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods, Instruments, & Computers,* vol. 28, no. 2, pp. 197-202, 1996.

[18] Y. Goldberg and O. Levy, *word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method,* arXiv preprint, 2014.

[19] T. K. Ho, "Random decision forests," *Proceedings of 3rd International Conference on Document Analysis and Recognition,* 1995.

[20] A. McCalluml and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 workshop on learning for text categorization,* vol. 752, pp. 41-48, 1998.

[21] C. D. Manning, P. Raghavan and H. Schutze, Text classification and Naive Bayes, Cambridge university press, 2008.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* p. 273–297, 1995.

[23] T. G. Dietterich, "Machine Learning Research: Four Current Directions," *AI Magazine,* vol. 18, pp. 97-136, 1997.

[24] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers,* vol. 10, no. 3, pp. 61-74, 1999.

[25] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* pp. 861-874, 2006.

[26] Scrapy, "scrapy/throttle.py at 129421c7e31b89b9b0f9c5f7d8ae59e47df36091 · scrapy/scrapy · GitHub," 27 June 2015. [Online]. Available:

https://github.com/scrapy/scrapy/blob/129421c7e31b89b9b0f9c5f7d8ae59e47df36091/scrapy/extensions/throttle.py. [Accessed 4 October 2017].

[27] Scrapinghub, "Scrapy | A Fast and Powerful Scraping and Web Crawling Framework," [Online]. Available: https://scrapy.org/. [Accessed 16 February 2017].

[28] X. Grangier, "GitHub - grangier/python-goose: Html Content / Article Extractor, web scrapping lib in Python," 29 3 2015. [Online]. Available: https://github.com/grangier/python-goose. [Accessed 15 4 2017].

[29] scikit-learn, "sklearn.decomposition.TruncatedSVD — scikit-learn 0.18.1 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html. [Accessed 15 April 2017].

[30] scikit-learn, "sklearn.naive_bayes.MultinomialNB — scikit-learn 0.18.1 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html. [Accessed 16 April 2017].

[31] BOR, "Roccurves.png," 9 January 2006. [Online]. Available: https://en.wikipedia.org/wiki/File:Roccurves.png. [Accessed 10 February 2017].

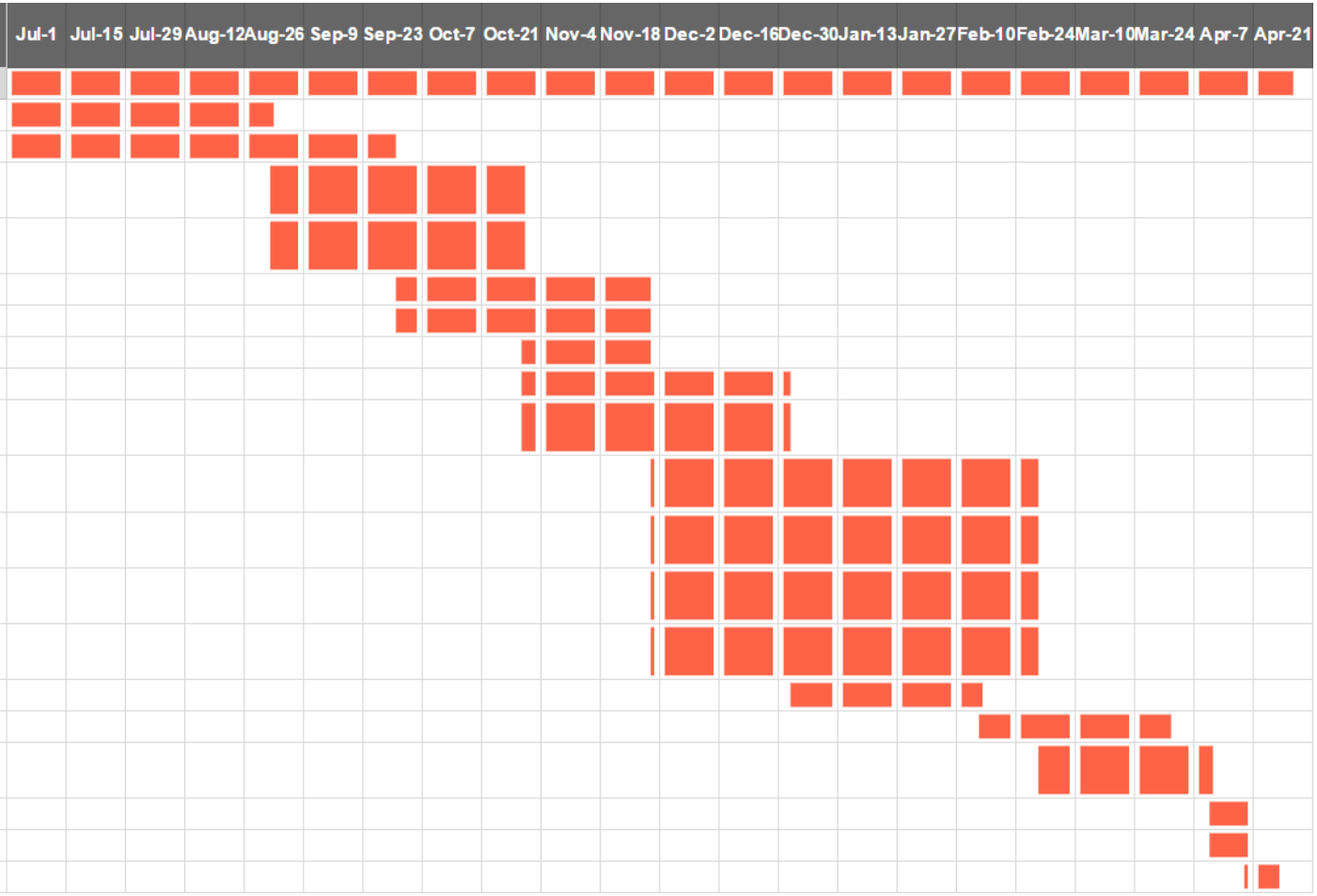
[32] S. Aseervatham, A. Antoniadis, E. Gaussier, M. Burlet and Y. Denneulin, "A sparse version of the ridge logistic regression for large-scale text categorization," *Pattern Recognition Letters,* vol. 32, no. 2, pp. 101-106, 2011.

6 Appendix

6.1 Appendix A – Project Planning

See next page for GANTT chart

Project	Start Date	End Date	Days
Task	**1-Jul**	**30-Apr**	303
Brainstorming	**1-Jul**	**1-Sep**	62
Literature Survey	**1-Jul**	**30-Sep**	91
Design and develop company URL retriever	**1-Sep**	**31-Oct**	60
Use FullContact Compnay API to fetch company information	**1-Sep**	**31-Oct**	60
Develop LinkedIn scraper	**1-Oct**	**30-Nov**	60
Develop company information scraper	**1-Oct**	**30-Nov**	60
Develop text pre-processor	**31-Oct**	**30-Nov**	30
Develop text feature extractor	**31-Oct**	**31-Dec**	61
Small scale test of the algorithm using random forest	**31-Oct**	**31-Dec**	61
Incorporate expert knowledge into scraper	**1-Dec**	**28-Feb**	89
Train and tune the models using random forest	**1-Dec**	**28-Feb**	89
Train and tune the model using Bernoulli naive Bayes	**1-Dec**	**28-Feb**	89
Train and tune the model with expert knowledge	**1-Dec**	**28-Feb**	89
Write the Progress Report	**1-Jan**	**15-Feb**	45
Tune the ensemble of models	**15-Feb**	**1-Apr**	45
Evaluating the system with partnering company	**1-Mar**	**10-Apr**	40
Write the final report	**10-Apr**	**19-Apr**	9
Design the project poster	**10-Apr**	**20-Apr**	10
Prepare for the presentation	**20-Apr**	**26-Apr**	6



6.2 Appendix B – Required Hardware & Software

6.2.1 Hardware Requirement

Development PC	Machine running Linux

6.2.2 Software Requirement

Python	Programming language
Vim	Code editor
CSV, JSON	Data format for communication
Scrapy, Goose	Web scraping library
SnowballStemmer	Text preprocessing library
nltk, scikit-learn	Machine learning library
matplotlib	Graph plotting
Google Sheets	Data processing and graph plotting
Git	Version control software

6.3 Appendix C – Meeting Minutes

6.3.1 Minutes of the 1st Project Meeting

Date	2016/09/13
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. No minute to approve
2. Report on progress
    i. Received sales data from partnering company
    ii. Completed company website URL crawler/retriever
    iii. Completed company social profile URL crawler/retriever
3. Discussion items
    i. Rate-limiting problem and accuracy of URL crawlers/retrievers
    ii. Need to make the URL crawlers/retrievers work with Chinese company names
4. Goals before next meeting
    i. Need to do more literature survey to ensure the thesis has based on proper technical background
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 4:05pm
    ii. The next meeting will be held on 10/18 3:30pm in Room 3512

6.3.2 Minutes of the 2nd Project Meeting

Date	2016/10/18
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Improved accuracy of company website URL crawler/retriever with Chinese company name
    ii. Circumvented DuckDuckGo's rate-limiting by auto-throttling
    iii. Attempted to fetch company information using FullContact Company API
    iv. Began building company information scraper
    v. Investigate the Python libraries that can be used to preprocess text
3. Discussion items
    i. Whether I should code the machine learning algorithms myself or I can feel free to use existing libraries
    ii. Price problems of FullContact Company API
    iii. Effectiveness of automatic web scraping
4. Goals before next meeting
    i. Need to improve the company website scraper
    ii. Should begin working on the next part of the project as soon as possible
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 3:55pm
    ii. The next meeting will be held on 11/15 3:30pm in Room 3512

6.3.3 Minutes of the 3rd Project Meeting

Date	2016/11/15
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Successfully scraped company information from company website and social profile
    ii. Able to remove non-English term and stop words; perform stemming from scraped text data
3. Discussion items
    i. Demonstrated how to automatically login to LinkedIn to perform scraping
    ii. Explained stemming and removal of non-English terms and stop words
4. Goals before next meeting
    i. Begin building predictive classifiers using existing data
    ii. Look at Python libraries that can be used to build the classifiers
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 4:00pm
    ii. The next meeting will be held on 12/06 3:30pm in Room 3512

6.3.4 Minutes of the 4th Project Meeting

Date	2016/12/06
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Successfully extracted term feature from company description using tf-idf and LSA
    ii. Performed small-scale test of training classifier using random forest
    iii. Calculated the precision, recall and F1 score of the outcome of the small-scale test
3. Discussion items
    i. Demonstrated the small-scale test
    ii. Presented precision, recall and F1 score distribution in histograms
    iii. Reflected on the precision, recall and F1 score and training data size
4. Goals before next meeting
    i. Tune to classifiers to make them more accurate
    ii. Attempt to keep similar good precision, recall and F1 score with larger data set
    iii. Try out other machine learning algorithms that can be used
    iv. Begin working on expert knowledge-based classifier
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 4:00pm
    ii. The next meeting will be held on 2017/02/06 3:30pm in Room 3512

6.3.5 Minutes of the 5th Project Meeting

Date	2017/02/06
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Developed scraper that scrapes according to expert knowledge requirement
    ii. Performed full-scale test on the data set using random forest, got F1 score of around 0.7
    iii. Trained Gaussian naïve Bayes model, got F1 score of around 0.6-0.69
3. Discussion items
    i. How to improve accuracy
    ii. How the data behave differently on different models
    iii. How to combine prediction results of multiple models
    iv. How imbalanced training data caused F1 score to drop to almost 0
4. Goals before next meeting
    i. Tune to classifiers to make them more accurate
    ii. Finish up expert knowledge classifier
    iii. Try out other machine learning algorithms that can be used
    iv. Begin working on expert knowledge-based classifier
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 4:05pm
    ii. The next meeting will be held on 2017/02/20 3:30pm in Room 3512

6.3.6 Minutes of the 6th Project Meeting

Date	2017/02/20
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Finished expert knowledge classifier using SVM
3. Discussion items
    i. How to improve accuracy
    ii. How expert knowledge classifier works
4. Goals before next meeting
    i. Tune to classifiers to make them more accurate
    ii. Work on combining prediction results
    iii. Begin using Word2vec to extract feature vectors
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 3:55pm
    ii. The next meeting will be held on 2017/03/06 3:30pm in Room 3512

6.3.7 Minutes of the 7th Project Meeting

Date	2017/03/06
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Finished training classifiers using feature vectors from Word2vec
    ii. Finished combining prediction results from different models
    iii. Presented precision, recall and F1 score distribution for before and after combining results from models
3. Discussion items
    i. Why naïve Bayes performs poorly using vectors from Word2vec
    ii. How to improve naïve Bayes performance using Word2vec vectors
4. Goals before next meeting
    i. Tune to classifiers to make them more accurate
    ii. Continue to improve the method to combine prediction results
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 3:55pm
    ii. The next meeting will be held on 2017/03/20 3:30pm in Room 3512

6.3.8 Minutes of the 8th Project Meeting

Date	2017/03/20
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

1. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
2. Report on progress
    i. Discovered the two-pass approach that works better than one-pass approach when combining predictions from different models
    ii. Modified LinkedIn scraper to adapt to the change in page structure after LinkedIn has updated it
3. Discussion items
    i. Explain why two-pass approach of combining predictions outperforms one-pass approach
    ii. Explain the updated approach to scrape LinkedIn
4. Goals before next meeting
    i. Tune to classifiers to make them more accurate
5. Meeting adjournment and next meeting
    i. The meeting was adjourned at 3:45pm
    ii. The next meeting will be held on 2017/04/03 3:30pm in Room 3512

6.3.9 Minutes of the 9th Project Meeting

Date	2017/04/03
Time	3:30pm
Place	Room 3512
Present	Ku Chun Kit, Dr. David Rossiter
Recorder	Ku Chun Kit

6. Approval of minutes
    i. The minutes of the last meeting were approved without amendment
7. Report on progress
    i. Discovered that K-nearest neighbors works much better than naïve Bayes when the feature vectors are extracted using Word2vec
    ii. Tried out different configuration of the models to optimize their accuracies
8. Discussion items
    i. Explain how different configuration options affected the accuracies
    ii. Explain why K-nearest neighbors works better than naïve Bayes
9. Goals before next meeting
    i. Tune to classifiers to make them more accurate
    ii. Begin writing final report
10. Meeting adjournment and next meeting
    i. The meeting was adjourned at 3:55pm

6.4 Appendix D – Sample Company Descriptions Retrieved

Table 6.1 Sample company descriptions retrieved

URL	Company description
http://www.aiesec.hk/about-aiesec/	AIESEC is a global platform for young people to explore and develop their leadership potential. We are a non-political, independent, not-for-profit organisation run by students and recent graduates of institutions of higher education. Its members are interested in world issues, leadership and management. AIESEC does not discriminate on the basis of ethnicity, gender, sexual orientation, religion or national / social origin. Since we were founded, we have engaged and developed over 1,000,000 young people who have been through an AIESEC experience. The impact of our organisation can be seen through our alumni who represent business, NGO and world leaders, including one Nobel Peace Prize laureate, Martti Atisaari of Finland.
http://www.hktdc.com/mis/ahktdc/en/s/abt-hktdc-about.html	The Hong Kong Trade Development Council (HKTDC) was established in 1966. As a statutory body, our mission is to create opportunities for Hong Kong companies, especially small and medium-sized enterprises (SMEs), by promoting trade in goods and services worldwide. With the help of our global network of more than 40 offices, including 13 on the Chinese mainland, we explore markets for Hong Kong SMEs and connect them with business partners around the world, while offering a variety of business-enabling services. Through a spectrum of activities, the HKTDC promotes Hong Kong as Asia's global business platform, reinforcing its reputation as Asia's premier services hub. We also provide a comprehensive array of trade-support seminars and other information channels to enhance Hong Kong SMEs' capabilities. Please visit www.hktdc.com for the latest information about the HKTDC and Hong Kong-related opportunities.
http://www.isoc.hk/about-us/	The vision of ISOC HK is that an open and accessible network, in technology and policy, contributes to and provides a platform for sustainable development of the information society, which enriches the human experience. The mission of ISOC HK is to: 1. Foster participation, contribution and leadership from individuals and organisations in Hong Kong, on the open development of standards, protocols, administration, governance and the technical infrastructure and evolution of the Internet,

	as local and global citizens of the information society; 2. Promote the Internet as a positive tool and environment for social cooperation, community building, and fostering of a culture that enables self-governance and balanced multi-stakeholder participation to work; 3. Be a voice and platform for Internet professionals, users and the Internet community at large in cooperative efforts on local and international policy, practice and development; 4. Provide and facilitate forums for idea, experience and cultural exchange among individuals and organisations from private and public sectors as well as across borders; 5. Support relevant educational, humanitarian and societal initiatives concerning the Internet community and the information society, such as addressing the bridging of digital divide, capacity and access building, as well as other informational and outreach activities. ISOC HK (Internet Society Hong Kong Chapter) is dedicated to the open, unencumbered, beneficial use of the Internet; the upholding of the freedom of expression and opinion, privacy of personal information and aversion of social discrimination; through responsible self-regulation and harmonized governance. Consistent with the ISOC statement of purpose, ISOC HK believes in ?馳aintain[ing] and extend[ing] the development and availability of the Internet and its associated technologies and applications ??both as an end in itself, and as a means of enabling organizations, professions, and individuals [locally and] worldwide to more effectively collaborate, cooperate, and innovate in their respective fields and interests.??ISOC HK will be driven by individual users and Internet professionals, in conjunction with and augmented by socially responsible corporations in the information society. ISOC HK intends to encourage participation and discussion form its members on issues within the scope of its mandate, and will endeavour to partner and join with other relevant groups from different spectrums of the local and global community on different activities. www.isoc.org The Internet Society (ISOC) is a nonprofit organisation founded in 1992 to provide leadership in Internet related standards, education, and policy. With offices in Washington, USA, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. The Internet Society provides leadership in addressing issues that confront the future of the Internet, and is the organisational home for the groups responsible for Internet infrastructure

	standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). The Internet Society acts not only as a global clearinghouse for Internet information and education but also as a facilitator and coordinator of Internet-related initiatives around the world. For over 15 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time. The Internet Society has more than 146? organisational and more than 78,000 individual members in over 110?chapters around the world. ISOC has also created 5 regional bureaus to better serve the regional Internet community. The Latin American and Caribbean bureau is located in Buenos Aires, Argentina, the African bureau in Addis Ababa, Ethiopia and the South and Southeast Asian bureau in Suva, Fiji. Through its sponsored events, developing-country training workshops, tutorials, public policy, and regional and local chapters, the Internet Society serves the needs of the growing global Internet community. From commerce to education to social issues, our goal is to enhance the availability and utility of the Internet on the widest possible scale. For more details about ISOC, please visit www.isoc.org
http://www.lkfgroup.com/index.php/about-us/	The Lan Kwai Fong Group is a dynamic company involved in diverse sectors in Hong Kong as well as major cities throughout Asia. The core of the business involves property acquisition, development and management for both residential & commercial use. Other businesses include high-end restaurants and bars, financial investments, movie production & distribution, community services and beverage production. Some of the successful properties under the management of LKF Group include the world famous Lan Kwai Fong Hong Kong restaurant & bar district, the luxurious Andara Resort & Villas in Phuket Thailand and Lan Kwai Fong Chengdu, Mall of the World and many more. Dr. Allan Zeman is the hands-on chairman behind Lan Kwai Fong Group. Having spent over 35 years in Hong Kong andChina, he has a keen understanding of what the market will want next. Under his meticulous eye, many projects and developments have shown tremendous success and growth, such as?Ocean Park in Hong Kong, Wynn Casino and Resort Macau, After Dark Film Production in Los Angeles and many more. The core of the business involves property acquisition, development and management for both commercial and residential use.

http://www.med.hku.hk/about-the-faculty	The Medical Faculty of The University of Hong Kong (HKU) is the longest established institution in higher education of Hong Kong. It was founded as the Hong Kong College of Medicine for Chinese by London Missionary Society in 1887, and was renamed as the Hong Kong College of Medicine in 1907. The Faculty was deemed as the premier Faculty when the University was established in 1911. Serving Hong Kong for over a century, it has firmly established itself as a medical school of learning, innovation, and enterprising; it is a medical school of moral, vision, and care. Since its inception, the Faculty has been playing a pioneering role in medical education, training and research. From its modest beginning, the Faculty has now become the largest faculty of the University, with over 400 full-time academic and academic-related staff and 800 research and research-related support personnel. The undergraduate student population is about 2,900 and the postgraduate student population is about 1,500. The Faculty is comprised of 14 departments, School of Biomedical Sciences, School of Chinese Medicine, School of Nursing, School of Public Health and a number of research centres focusing on various strengths of research.
http://www.scmp.com/business/companies	After Starbucks launched its latest mobile ordering feature in Hong Kong earlier this month, its executive director believes the next step for city coffee fans is to have their Starbucks order delivered straight to their doorstep.

6.5 Appendix E – Sample Social Profile URLs Retrieved

Table 6.2 Sample Crunchbase profiles retrieved

Internet Society Hong Kong	https://www.crunchbase.com/organization/uk-broadband-limited
Accuvally	https://www.crunchbase.com/organization/crunchbase
AIESEC HONG KONG	https://www.crunchbase.com/organization/chinese-university-of-hong-kong
Alibaba.com Hong Kong Limited	https://www.crunchbase.com/organization/alibaba
AngelHack	https://www.crunchbase.com/organization/angelhack
APAC BioHealth Consulting	https://www.crunchbase.com/organization/asia-pacific-internet-group
AQ Communications Limited	https://www.crunchbase.com/organization/azzurri-communications
Arrow Asia	https://www.crunchbase.com/organization/arrow-asia-pac
Balenciaga	https://www.crunchbase.com/organization/lyst
Bentley Communications Ltd.	https://www.crunchbase.com/organization/avanti-communications

Table 6.3 Sample Facebook profiles retrieved

Internet Society Hong Kong	https://www.facebook.com/ISOCHK
Accuvally	(no result)
AIESEC HONG KONG	https://www.facebook.com/AIESECHongKong
Alibaba.com Hong Kong Limited	https://www.facebook.com/pages/Alibabacom-Hong-Kong/546468715385599
AngelHack	https://www.facebook.com/AngelHackHK
APAC BioHealth Consulting	https://www.facebook.com/apacbiohealthconsulting/
Apprendre Education Limited	https://www.facebook.com/pages/Elite-Education-Learning-Centre-Limited/114459591924796
AQ Communications Limited	https://www.facebook.com/aqcomm
Arrow Asia	https://www.facebook.com/arrowasiapac

Balenciaga	https://www.facebook.com/Balenciaga
Bentley Communications Ltd.	https://www.facebook.com/BentleyHaulageLtd

Table 6.4 Sample LinkedIn profiles retrieved

Internet Society Hong Kong	https://www.linkedin.com/company/internet-society-hong-kong
Accuvally	https://www.linkedin.com/company/accuvally-inc-
AIESEC HONG KONG	https://www.linkedin.com/company/aiesec-in-hong-kong
Alibaba.com Hong Kong Limited	(no result)
AngelHack	https://www.linkedin.com/company/angelhack
APAC BioHealth Consulting	https://www.linkedin.com/company/apac-consulting-and-information-services
Apprendre Education Limited	https://www.linkedin.com/company/eurasia-fund-management-limited
AQ Communications Limited	https://www.linkedin.com/company/aq-communications-limited
Arrow Asia	https://www.linkedin.com/company/arrow-asia-pac-ltd
Balenciaga	https://www.linkedin.com/company/balenciaga
Bentley Communications Ltd.	https://www.linkedin.com/company/bentley-communications

6.6 Appendix F – Qualified Industries According to Legacy Sales data

Table 6.5 Industries that are considered qualified in expert knowledge qualifier

Airlines/Aviation	International Trade and Development
Apparel & Fashion	Internet
Banking	Legal Services
Civic & Social Organization	Luxury Goods & Jewelry
Commercial Real Estate	Management Consulting
Computer Software	Marketing and Advertising
Consumer Goods	Media Production
Design	Newspapers
E-Learning	Nonprofit Organization Management
Education Management	Philanthropy
Electrical/Electronic Manufacturing	Primary/Secondary Education
Entertainment	Public Relations and Communications
Events Services	Real Estate
Financial Services	Research
Fine Art	Retail
Government Administration	Semiconductors
Government Relations	Sports
Graphic Design	Staffing and Recruiting
Hospitality	Telecommunications
Human Resources	Transportation/Trucking/Railroad
Import and Export	Utilities
Information Services	Venture Capital & Private Equity
Information Technology and Services	Wine and Spirits

6.7 Appendix G – Detailed Description of LinkedIn Scraping

Before LinkedIn's layout change on March 2017, scraping LinkedIn was done by looking up for specific HTML element with a specific id attribute from the source HTML of the page. However, since the change on March 2017, request to LinkedIn company profile would return a 302 redirect:

```
[scrapy.downloadermiddlewares.redirect] DEBUG: Redirecting (302) to <GET
https://www.linkedin.com/company-beta/2818799?pathWildcard=2818799> from <GET
https://www.linkedin.com/company/exhibition-services-ltd>
```

This means the URLs to the old layout are being mapped to the new layout and users would be redirected to see the new layout when accessing LinkedIn.

After March update, LinkedIn's page structured has been updated to this format:

```
<html>
  <head>
    <!-- Some meta tags -->
  </head>
  <body>
    <code style="display:none;" id="datalet-bpr-guid-1234">
      {"request": "/voyager/api/something/company_id=123456", status: 200, body:
"bpr-guid-1234"}
    </code>
    <code style="display:none;" id="bpr-guid-1234">
      {"data": {"data":"here","more_data":"in this json object"}}
    </code>
    <!-- some more code tags similar to the two above -->
  </body>
</html>
```

At first glance, there is no explicit visible HTML element that can be taken for data directly because there is virtually no visible HTML element. All HTML code elements were given an CSS style “display: none” that prevents the codes from being shown.

Also, the pair of “`bpr-guid-1234`” and “`datalet-bpr-guid-1234`” element are curious as one looks like an internal API call within LinkedIn and another looks like the response of it. I assume that LinkedIn web server backend has a number of API services so different LinkedIn applications (mobile app/web) can share the same set of APIs.

Since each API call like this

`/voyager/api/something/company_id=123456`

is unique inside one LinkedIn company profile page, by doing regular expression search on “`/voyager/api/something/company_id=`” it was possible for to obtain the unique “`bpr-guid-1234`” id of the element that contains the data we need.

6.8 Appendix H – Organization of Source Codes

The source codes of web crawler, web scraper, text preprocessors and classifiers for this thesis are managed using version control software Git. The development PC maintains a copy of the git repository while the remote server hosted by GitLab (https://gitlab.com) maintain another copy of the repository. Using version control can ensure that in the event of accidental deletion of source codes, only the latest changes made to the codes will be affected as a backup copy is always available locally or on a remote server. Version control also makes it easy to revert to previous versions of the program in case a new change breaks the program.

The sources codes are organized as three separate git projects:

1. url-scraper
   i. Company website URL crawler
   ii. Company social profile URL crawler
2. info-scraper
   i. Company website scraper
   ii. LinkedIn scraper
   iii. Expert knowledge scraper for company website
   iv. Expert knowledge scraper for LinkedIn profile
3. tfidflsa
   i. Text preprocessors
   ii. Classifiers using tf-idf + LSA
   iii. Classifiers using Word2vec

iv. Auxiliary scripts as experiments