NORTHWESTERN UNIVERSITY

Topics in Document Classification

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Industrial Engineering and Management Sciences

By

Papis Wongchaisuwat

EVANSTON, ILLINOIS

March 2018

Abstract

Unstructured data like text is plentiful and possibly contains valuable insights leading to a better decision-making process. Manually obtaining these insights can be costly and time-consuming. Text mining, also known as Text analytics, is developed to derive meaningful information from textual data. It is widely applied in various domains such as business-oriented problems, legal space, social media, and biomedical applications. This dissertation aims to apply text mining techniques including linguistic information retrieval, statistical and machine learning to solve three different problems.

Patent litigations are generally unpredictable, disruptive, and expensive. An ability to predict the patent likelihood and estimate time to litigation in advanced is profitable in many aspects. We propose predictive models relying on textual and non-textual features to forecast patent litigations and time to litigation in the second chapter. In the next chapter, we consider an application of text mining techniques in the health-care domain. In Community-based Question Answering sites, several health-related questions are posted but remain unanswered. We consequently develop an automate system to answer questions based on past question-answer pairs. We address a semantic aspect of textual statements in the last chapter. Contents from various sources especially from web sites are not necessarily reliable which potentially cause negative impacts to readers. Hence, an algorithm to validate the truthfulness of statements and provide supporting evidence for a false triplet is proposed.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Diego Klabjan, for his continuous support and mentorship throughout my Ph.D. study. His guidance with motivation and patience was paramount in advancing my research skills. I would like to thank Professor Siddhartha R Jonnalagadda for his supervision in the third chapter of my dissertation. Besides my advisors, I thank Professor Sanjay Mehrotra for his insightful comments and advice as my prospectus and thesis committee member.

I acknowledge and extend my thanks to the Ananda Mahidol Foundation for the financial support throughout my graduate studies. As a fellowship receiver, I would like to express my highest gratitude to His Majesty King Bhumibol Adulyahdej of Thailand, who established the foundation to support Thai students for pursuing their higher education abroad.

I am thankful to all professors, staff members, and colleagues at the IEMS department, Northwestern University. Their support makes my graduate school experience in these part five years become invaluable and memorable. Additionally, all my friends both in the US and in Thailand earn my thankfulness for their suggestions and encouragement during my difficult times. Specifically, I thank my best friend, Saisattha (Buahom) Noomnual for sharing good and hard times together. I also send my thanks to Pakpoom (Pai) Buabthong for his support in all aspects.

Finally, and more importantly, I could not have come this far without the love and endless support from my dad, mom, and sister. They truly understand and constantly give me unconditionally encouragement. I send my love and gratitude to my family.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Textual data is more difficult to be quantified compared to well-structured data. Text mining techniques are developed and widely used to extract valuable information from textual data. These techniques are broadly applicable for a variety of needs both in academic research and business aspect. This dissertation is concerned with three different topics in document classification problems; 1) Predicting patent litigations; 2) A Semi-Supervised Learning Approach to Enhance Community-based Question Answering and 3) Truth validation and Evidence. These topics are extensively discussed in chapter 2, 3 and 4, respectively.

In chapter 2, we develop a predictive model for forecasting a likelihood of a patent litigation. For litigated patents, a model to estimate an expected time to litigation is proposed. Our work focuses on improving the state-of-the-art by relying on a different set of features and employing more sophisticated algorithms with more realistic data. We consider potential factors influencing a likelihood of a patent litigation in the model. These features, collected at the issue date of the patent and thus prior to the actual litigation, include textual features, patent's general information as well as financial information of patent's assignee. Our proposed models are a combination of a clustering approach coupled with an ensemble classification method. The initial model for predicting the likelihood is further modified to capture a time-to-litigation perspective. With a low litigation rate of 1 to 2 percent in practice, results from the models show promising

predictability. Financial information and features related to referencing are important indicators to distinguish between litigated and non-litigated patents.

The next chapter considers Community-based Question Answering (CQA) sites, such as Yahoo! Answers, that play an important role in addressing information needs. In most CQA sites, a significant number of posted questions remain unanswered. In this study, we hence develop an algorithm to automatically answer questions based on past questions and answers (QA). Our proposed algorithm uses information retrieval techniques to identify candidate answers from resolved QA. These candidates are further ranked by using a semi-supervised leaning algorithm. We assess this approach on a curated corpus related to alcoholism as a case study and compare it against a rule-based string similarity baseline. Our automated QA system based on historical QA pairs is shown to be effective according to the data set in this case study. Important features distinguishing a valid answer from an invalid answer include text length, number of stop words contained in a test question, a distance between the test question and other questions in the corpus, and a number of overlapping health-related terms between questions.

Plentiful of textual data obtained from various sources especially web pages is not fully reliable. Misleading information potentially leads to disastrous consequences. Verifying statements manually is also costly and time-consuming. In the last chapter, we hence develop an automated system to verify the truthfulness of a statement and provide supporting evidence. Our main contribution is a novel algorithm to provide supporting evidence after the triplet has been identified as false. Our proposed algorithm relies on knowledge from reliable sources including a knowledge graph and ontologies. We employ an inference method based on the knowledge graph to validate the truthfulness of the statement. In order to provide supporting evidence for a false

triplet, we first matches entities from the knowledge graph with concepts from on ontologies. We construct a collection of evidence from specific concepts in ontologies. Then, a collection of evidence is summarized to be as concise as possible.

Chapter 2
Predicting Patent Litigations

2.1 Introduction

A litigation is commonly associated with high cost of patent lawsuits and time-consuming legal processes. Patents are means to protect intellectual property and they establish inventions. Some large companies including Canon, Google, IBM, Microsoft and Samsung have been obtaining and accumulating a large number of patents each year, especially in recent years[1]. Companies typically invest significant resources in acquiring and developing patents which can protect potential lawsuits in order to build a defensive-patent portfolio. There is uncertainty and difficulty to estimate which patents are likely to be litigated which leads to uncertainty in financial planning, such as how to allocate the Research and Development budget. Patent trolls who accumulate third party patents, instead of investing a large amount of money to collect a portfolio of patents, could improve their portfolio selection by having the ability to accurately indicate whether a patent has a high chance of being contested. The patent trolls could also take advantage of time-to-litigation predictions by better forecasting an exact time to purchase a patent. This work develops predictive models to differentiate between likely-to-be litigated and not-to-be-litigated patents ahead of time as well as to predict when the litigation is going to take place. Hence, the models developed in our

[1] Center for global innovation/patent metrics, Global patent quality statistics & investment analysis http://www.bustpatents.com/statistics.htm

work can help companies achieve a more realistic budget allocation plan or improve patent portfolios of patent trolls.

We develop a combination of clustering and classification models to predict which patent is likely to be litigated based on multiple features including textual types, which are extracted directly from the claim section in a patent, as well as non-textual types, which are obtained from other relevant information. The data set used in our work is highly imbalanced between litigated and non-litigated cases hence a re-sampling method is implemented prior to fitting the classification models. In the second part of the paper, we aim to forecast the time to litigation of disputed patents. The models are tested on different data sets including patent classes used by the USPTO associated with the following three keywords: "Wireless Network," "Advertising" and "Telecommunication." These selected keywords are related to the technology industry, and, unsurprisingly, technology companies hold large portfolios of patents. These companies commonly compete with each other, which can lead to a controversy regarding the ownership of an invention. Consequently, technology companies have a higher chance of being involved in patent lawsuits; we have therefore specifically selected technology-related keywords to test the models.

We develop the models based on three options of auxiliary data sets augmenting other features with different financial information (obtained from the U.S. Securities and Exchange Commission - SEC data). The three data options include the model without the SEC data, with the SEC data by eliminating records without the SEC information, and with the SEC data under assumptions regarding the default values for missing observations. In order to measure the performance of the model, we use common metrics including the precision and recall, the F1-

score, and the confusion matrix [1]. The overall F1-score is used to evaluate each class in the time-to-litigation model since this model is multiclass.

Given a test patent to be predicted, our models use a clustering approach combined with a heuristic technique as well as an ensemble classification method to predict the litigation likelihood and the time to litigation. In other words, the models specify whether the test patent belongs to the "litigate" class in the litigation model. For the time-to-litigation model, we categorize all cases into different year groups based on their time to litigation information. We apply a similar algorithm as the litigation model to each year group and finally re-adjust the predicted classes by taking the hierarchical relationship among year groups into consideration. For example, if a patent is predicted to be litigated before year 4, then it is also predicted to be litigated before year 5. Our algorithm relies on an anomaly detection idea which shares the same characteristic of identifying very rare events. The K-means algorithm is implemented as the baseline clustering approach for both models. The distance between the test case and convex hulls of the clusters are computed to determine which class it belongs to. The heuristic technique as well as the ensemble classification method between Support Vector Machine (SVM) and random forest are further used to re-estimate the class of the test case when the designation is too ambiguous from clustering.

Based on our study, the "Wireless Network" keyword with SEC data assuming no default value yields the highest F1-score of 0.19 for the litigation model. Among the three keywords, "Wireless Network" tends to give the highest F1-score, followed by "Telecommunication" and "Advertising" performs the worst. Using SEC information without a default value yields the best result while assuming a default value performs better than excluding SEC data. For the time-to-litigation model, "Wireless Network" generally yields a higher F1-score for most classes and data

options than the other two keywords. We observe that enhancing the classification method with clustering does not improve the time-to-litigation model's performance compared to utilizing only the classification model.

The features, the models, and the data sets used to test the models distinguish our work from others. Specifically, our work has three main contributions. First, we explore other informative features that have not been studied in prior works. We include number of referenced patents that were litigated, the second layer of references as well as PageRank score features in the models to capture deeper knowledge of referencing. We also introduce financial information of the patent's assignee into the models by including SEC data features. To our knowledge, no prior research exists implementing textual features to predict the litigation likelihood as well as time-to-litigation of a patent. Second, we use a combination of clustering methods adapted from anomaly detection models enhanced with a standard classification approach in the litigation model. We cluster litigated and non-litigated models and use convex hulls of clusters which has not yet been done in the past. Finally, we test the performance of all models with the testing data that reflects an actual rate of litigations, i.e. the severe imbalance of classes.

In Section 2.2, we discuss a literature review in detail. We describe our models thoroughly including features used in Section 2.3. Further description regarding data collection is provided in Section 2.4. The results of our models and further discussions are reported in Section 2.5 and 2.6. Conclusions and future work are stated in Section 2.7.

2.2 Relevant work

A patent is an intellectual right granted to an inventor in order to prevent others from using the same invention. Each year a large number of patents are granted by USPTO, and some of them are infringed. This inevitably leads to litigations. Chien [2] studied various factors influencing the likelihood of a patent litigation. Instead of focusing on intrinsic factors which are embedded within a patent, acquired characteristics developed after the patent has been issued were specifically analyzed. These acquired features include changes in patent ownership, continued investment in the patent, securitization of the patent, and citations to the patent. In the experimental data set, each randomly selected litigated patent was combined with three additional patents issued in the same year and assigned to the same class. The results indicated that patents having a higher chance of getting disputed can be distinguished in advance from those being less likely to be involved in a dispute. Chien reported the predictability of the model by comparing the number of patents predicted to be litigated versus the number of actually litigated patents. Although the performance reported in [2] is better than ours, the litigation ratio assumed in [2] is much higher. Although the author's work undoubtedly made a significant contribution to the field of predicting patent litigations, there are weak points that we address in the work herein. The data experimented in Chien's model was designed arbitrarily by creating a matched-pair set from the same technology class, which does not represent the actual rate of litigation. More importantly, Chien's model could potentially be impractical as it relies on many features that can only be obtained after the litigation starts.

Petherbridge [3] pinpointed the limitation of Chien's model in terms of accuracy and practicability including a high false positive rate. Comparing with our work, Petherbridge provided

an explanation illustrating a relatively rare event of the patent litigation problem while we tested the models with actual data. Even though our model shows the similar issue as Chien's work of a high misclassification potential, it can be calibrated to reduce the false positive rate under an acceptable false negative rate. Kesan et al. [4] provided follow-up research to Chien's work where multiple flaws and possible improvements were discussed including the data collection, the features, and the normative conclusions. In summary, these works discussed limitations of Chien's work in both methodological and usage perspectives. They provided some possible solutions without implementing the actual models. We address some of these flaws. In particular, our models use actual data covering a larger time span as pinpointed by [3] and [4] . We introduce different informative features that have not yet been considered in prior works such as referenced knowledge, financial information features as well as textual features. Textual features are collected from patent claims which are import factors closely related to resolving complex legal situations [5]. These features can be gathered when the patents are issued, which addresses the issue stated in [4] that some features used by Chien such as whether the patent was reassigned and whether the patent was in reexamination cannot be obtained at the time of the prediction. It is likely that the patent is often reassigned only after realizing the litigation's decision [4] .

Su [6] compared characteristics of litigated with that of non-litigated based in order to understand the differences between these two types. A statistical test and descriptive statistics for patent characteristics among litigated and non-litigated patents were considered. A logistic regression was conducted to estimate the probability of a patent litigation. Specifically, a logistic function was used for the curve-fitting based on the whole data without separating to training and testing purposes. No model evaluation was taken into a consideration. Moreover, empirical factors

determining which patent is more likely to be litigated were studied in [7] and [8]. According to these studies, both information on patent lawsuits and patent documents were collected and combined to identify how the characteristics of patents affect the likelihood of litigation. Multiple hypotheses were statistically tested. The number of claims, the technology-based classification system of the patent, citations, country and type of ownership, and the size of the patent portfolio were considered as factors in these papers.

An empirical analysis of determinants of a patent litigation in a German court was studied in [9]. Similar results based on the US legal system were obtained, except that there was no difference between the chance of facing litigation among individual patent owners and companies in the German system. Unlike our work, which implements machine-learning algorithms as predictive models, [6], [7], [8] and [9] mainly focused on studying determinants of patent lawsuits rather than their prediction. Considering features used in our model, number of inventor, number of claim, number of backward reference, number of foreign reference features are taken from these prior work while number of litigated backward reference, number of 2nd layer of reference, number of 2nd layer of litigated reference, average pagerank score of backward reference, and financial information features are newly introduced in our model. Comparing with [6], our work employs more sophisticated machine learning models to predict the probability of a litigated patents. Different evaluation metrics are used to measure the predictability of our model with a cross-validation technique implemented.

Another interesting line of research is predicting the outcome of a litigation especially a patent litigation. Ashley and Bruninghaus [10] automatically classified textual descriptions of the facts of legal problem, which leads to an evaluation and prediction of the case scenarios' outcomes. Cowart et al. [11] implemented a logistic regression model and a classification tree to predict a

legal decision making process. In the study, both algorithms provided a similar overall prediction rate of 78 percent. Kashima et al. [12] developed a model predicting patent quality by measuring the stability of a patent, which is indicated by the possibility of patent surviving in court. The work employed a machine learning technique to predict the outcome of the IP High Court in Japan. These techniques include the Support Vector Machine (SVM), class-proportionate weighting scheme for the imbalanced data set, and the L1-regularization method for preventing over-fitting of the high-dimensional data set. The model proposed by Kashima et al. relied on both textual and non-textual (tailored) features. The pattern-based model consisting of bi- and tri-gram patterns as textual features yields the best performance with a 0.356 break-even point and 0.65 Area Under the Curve (AUC)[2] compared to word-based and tailored-based models. It focused on predicting a patent quality or the court outcome once the litigation started while ours aim to predict the litigation likelihood. The evaluation measures are also different as AUC used in [12] is not suitable for a highly imbalanced data set like ours.

Our problem is also related to the anomaly detection task where the number of anomalous items is relatively small compared to the whole data set. In our case, a litigation (anomaly) is a rare event. Three main categories of anomaly detection techniques including supervised, semi-supervised and unsupervised are reviewed in [13]. With availability of labeled data, a predictive model for normal and anomalous classes is typically constructed. We instead use the clustering technique, a common unsupervised approach, to enhance the performance of the classification model due to the lack of a large number of historical records, i.e., patents. To the best of our

[2] The break-even point evaluates a prediction accuracy using the optimal thresholding while AUC evaluates the goodness of ranking of instances given by the model. AUC is the expected proportion of a randomly-picked valid patent ranked higher than a randomly-picked invalid patent.

knowledge, we also contribute in this space since we cluster both classes and combine clustering with classification.

2.3 Methodology

Our goals are to predict which patent is likely to be disputed and when it would occur. We construct two main models, including the litigation likelihood model and the time-to-litigation model. The clustering approach combined with a variation of the nearest convex hull classification [14] is improved with the ensemble classification model. To address the issue of imbalanced data sets in training the classification models, we re-sample the data set. We use the same set of features in both litigated and time-to-litigation models, which are discussed in detail next.

2.3.1 Features

Features used in the models are divided into two distinct groups; 1) *textual features* and 2) *non-textual features.*

1. Textual features

We rely on the assumption that words occurring in a patent contain significant information in determining the litigation likelihood. Each patent consists of detailed information including the claim section from where we extract textual features. A document-term matrix is constructed based on all claims mentioned in the patent by incorporating unigram (one word), bigram (a pair of words) and trigram (a triple of words) features. The values in the matrix correspond to the term frequency-inverse document frequency (tf-idf) factor. The tf-idf value for a particular word increases if it appears frequently in the document but decreases when it relatively appears often in

all documents. This idea takes into account the fact that some common words appearing frequently in general should not be important. The generated matrix is large and thus we select 30 textual features with the highest information gain to be the final set of textual features.

2. Non-Textual features

In addition to the knowledge embedded in the claim section of a patent, other relevant information should be considered. These features which are extracted from the patent document and SEC websites are listed in Table 2.1. Financial data is discounted to the current time and categorized into groups. All features 1-9 are numerical values while features 10-12 are categorical values.

The 2nd layer of reference conveys more insights on how a patent relates to others, i.e. indicating patents within a similar area of interest. The PageRank score has been developed by Google to rank websites in the search engine results. It measures the significance of each web page based on other web pages linking to it. Applying this idea to our framework, we implement the PageRank score with the reference network constructed from patent documents. In this network, each node corresponds to a patent and there is an edge if and only if there is a reference relationship between the two patents. The PageRank score of each node is a weighted average of PageRank scores of all connected nodes (and thus defined recursively). The weight is assigned based on the significance of each node, i.e. a reciprocal of the number of outgoing edges. The basic idea behind this concept is that the more important a patent is, the more likely it receives links referenced from other patents.

The likelihood of litigation is potentially related to the business aspects of the company owning the patent. Three features collected from the SEC website including revenue, earnings per share and market share price represent the financial information of the patent's assignee.

Table 2.1 A list of features included in the model and their source of information

Non-textual features	Sources
General information about the patent	
1. Number of inventors	The patent document
2. Number of claims	The patent document
3. Number of words in claims	The patent document
4. Number of foreign references	The patent document
5. Number of backward references	The reference network constructed from patent documents
6. Number of 2nd layer of backward references	The reference network constructed from patent documents
7. Number of litigated backward references	The reference network constructed from patent documents
8. Number of 2nd layer of litigated backward references	The reference network constructed from patent documents
9. Average of PageRank score of backward references	The reference network constructed from patent documents
Financial data of the patent's assignee	
10. Revenue	The SEC website
11. Earnings per share	The SEC website
12. Market share price	The SEC website

2.3.2 The litigation model

Our first attempt to predict the litigation likelihood is utilizing the standard classification approach where the supervised learning algorithms of SVM, decision tree, boosted tree, random forest, as well as ensemble methods among these algorithms are experimented with. We call this approach

pure classification. The ensemble model between SVM and random performs best. The classification method performs satisfactorily but not as well as the following algorithm named *the cluster with ensemble method*.

We first cluster litigated and non-litigated cases in the training set by using the K-means algorithm as depicted in Figure 2.1. The K-means algorithm is an unsupervised learning approach which aims to categorize all records in the data set into a pre-defined number of clusters (k clusters). In each iteration, each record is assigned to its nearest centroid. After all points are assigned, *k* new centroids are re-calculated. These steps are repeated until no further changes can be observed. In what follows, we treat 1/0 as infinity applied to all ratio computations. The flow diagram of the cluster with ensemble algorithm is shown in Figure 2.2.

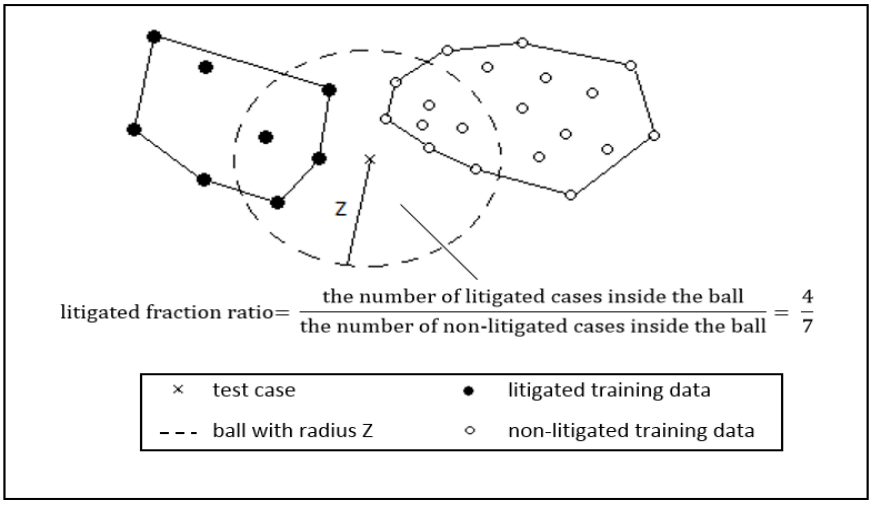

Figure 2.1 The cluster with ensemble approach

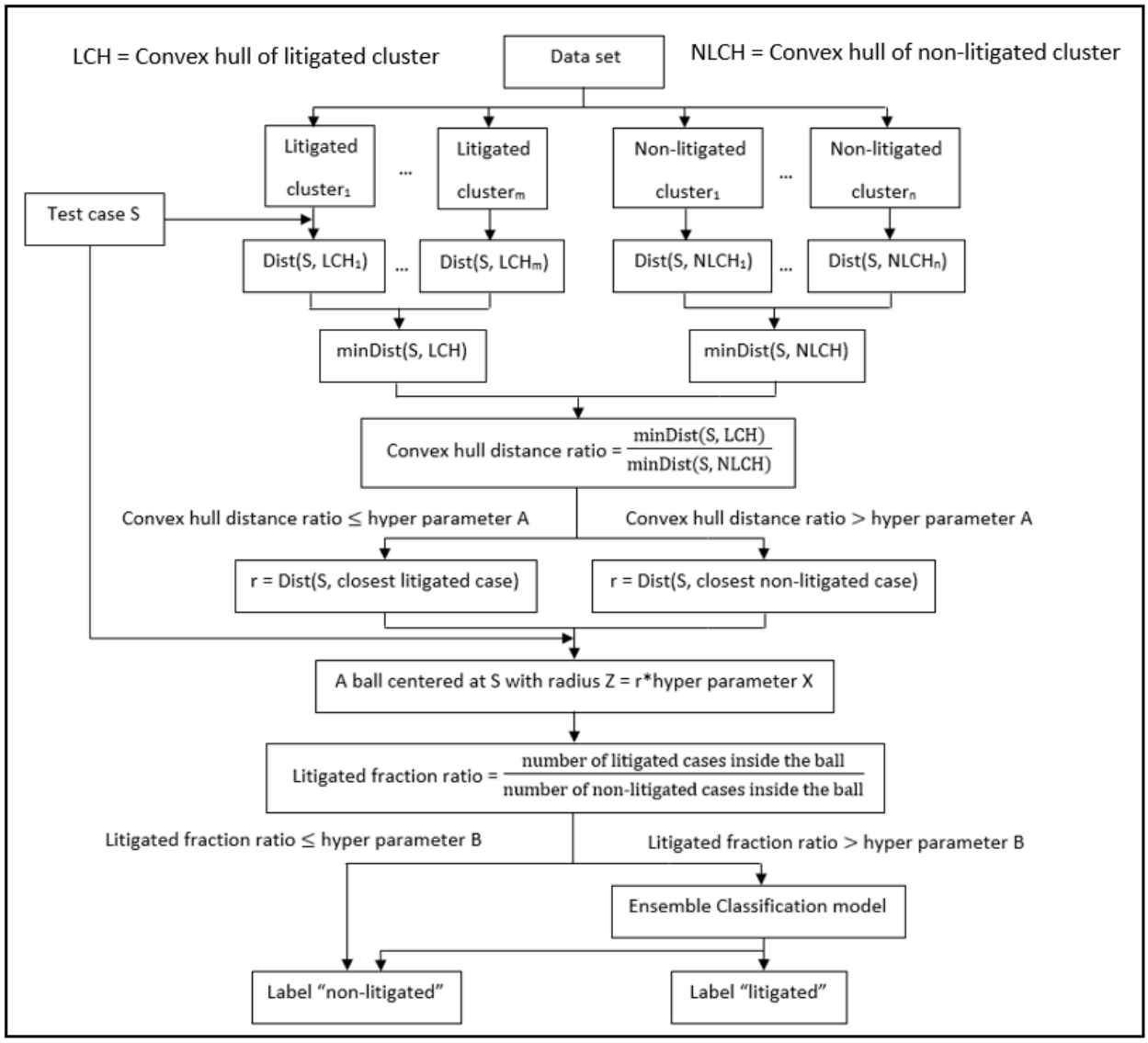

Figure 2.2 A flow diagram of the cluster with ensemble algorithm

In scoring, first, the distance between a test case and the convex hull constructed with all members in each cluster is computed. The ratio of the distance between the test case to the closest litigated and the closest non-litigated cluster, named convex hull distance ratio, is computed. The test case is initially assigned to be litigated if this ratio is smaller than some hyper parameter A, and non-litigated otherwise. We next construct a ball centered at the test case with radius z which

is a fraction of hyper parameter X and the distance r between the test case and the closest point of the class to which the test case was previously assigned to. We next compute the ratio between the number of litigated and non-litigated cases falling inside the ball, named litigated fraction ratio. If the ratio is lower than some hyper parameter B, the initial litigated label is assigned to be the non-litigated class while the initial non-litigated label remains the same. The ensemble classification model between SVM and random forest is applied to re-adjust the label when the litigated fraction ratio is greater than hyper parameter B.

2.3.3 The time-to-litigation model

The goal of this model is to predict the time to litigation for each disputed case. After collecting the number of years between a patent's issue date and its first litigation date, we categorize all cases into different groups including litigation before 14 years, 7 years, 4 years and 1 year after the issue date of the patent. We set cut-off points between groups by considering big differences in the histogram of time to litigation information in the data set. We use the time-to-litigation groups as the label to fit the models. Then, the final adjustment of a predicted class is implemented by considering that if a patent is predicted to be litigated by year 1, it definitely has to be litigated in later years (by year 4 or 7 or 14). The hierarchical tree indicating time to litigation of the model is illustrated in Figure 2.3.

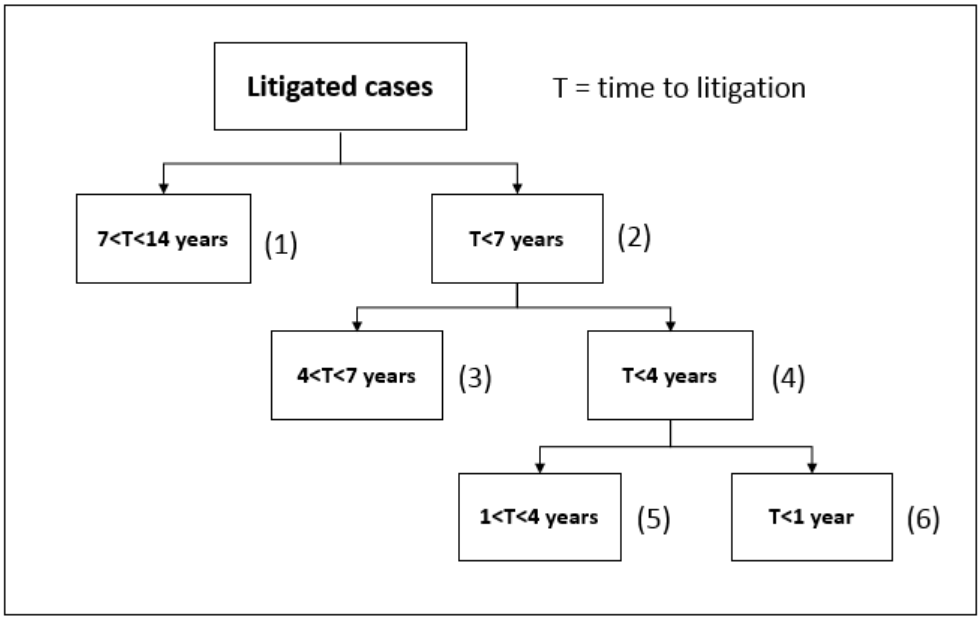

Figure 2.3 A hierarchical tree for the time-to-litigation model

In addition to fitting a model for each class independently, we also attempt to simultaneously fit a model for all classes while taking the hierarchy of classes into consideration. For the training part, we first cluster the final leaf node in the hierarchical tree (T<1 year) using the K-means algorithm and further expand the clusters with other classes in the hierarchical tree as depicted in Figure 2.4. We assign each case from the 1<T<4 years class to the closest T<1 cluster depending on the distance to the convex hull of the clusters. We repeat the process for the remaining classes until we achieve 4 layers of classes.

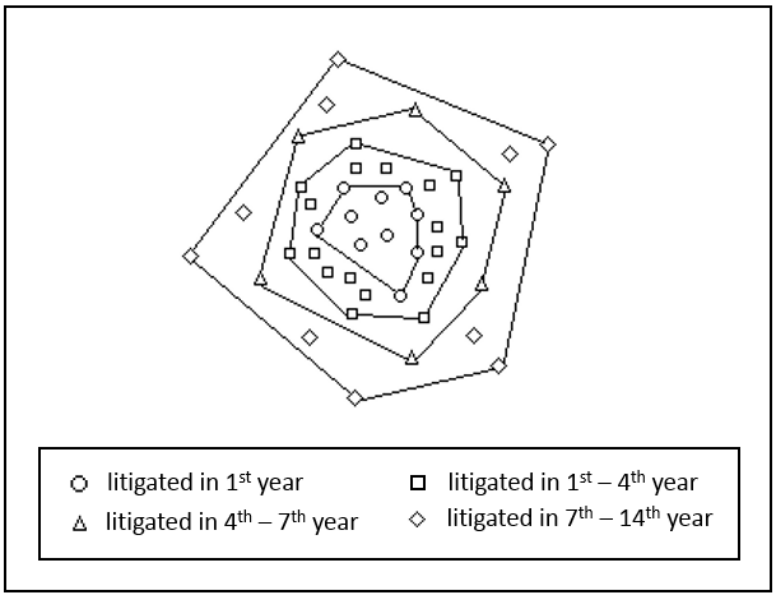

Figure 2.4 Layers of convex hulls of litigated cases

The time-to-litigation class is assigned based on which layer of the convex hulls each test case falls into. The test case is labeled with the closest convex hull it falls inside. For instance, the test case is classified as the T<4 year class if it falls inside the convex hull of the T<4 year class and outside the convex hull of the T<1 year class. If the test case falls outside the convex hull of T<14 years, it is labeled as T<14 years.

2.3.4 Re-sampling

The litigation rate is low with a value of 1 to 2 percent among all granted patents. With such an imbalanced data set, it is challenging for an algorithm to perform well. In order to enhance the predictive power of the classification model, the re-sampling technique is used to reduce the unbalancing level in the original data. Specifically, Synthetic Minority Over-sampling Technique (SMOTE) [15], which oversamples the minority class and undersamples the majority class is employed. In order to over-sample, a random point along the line segment between the minority

class sample and its k nearest neighbors is created. For under-sampling, samples are randomly removed from the majority class. For the litigation model, the minority and the majority classes used in SMOTE are litigated and non-litigated cases, respectively. We implement SMOTE technique for each label separately for the time-to-litigation model. For example, the minority class for the label "litigated by year 1" (node 6 in Figure 2.3) is the litigated cases which were only disputed by year 1, while the majority class includes other litigated cases which were disputed after year 1.

2.3.5 Model calibration and evaluation

After testing the fitted models, we achieve the predicted label directly from the clustering approach and the predicted probability belonging to each class from the classification model. To evaluate the litigation model's performance, traditional metrics including the confusion matrix, precision and recall as well as the F1-score are computed. The F1-score which is the geometric mean between the precision and recall values is used to compare different models. For the time-to-litigation model, the F1-score is computed for each class including nodes 2, 4 and 6 in Figure 2.3.

Among the different machine learning algorithms, SVM and random forest provide satisfactory performance. We consequently implement an ensemble method between these two algorithms by varying different weights given to each algorithm. The ensemble method is further used in the clustering approach which is also calibrated with different hyper parameters to maximize the model's performance. We ran all experiments with multiple replications of 10-fold cross validation to ensure consistency of the results. In other words, we split all records into 10 parts. We select 9 parts for training the model and test it on the part we leave out. We iterate this procedure by selecting different training (9 parts) and test (1 part) set. The evaluation metrics are

reported as the average across all 10 folds. Cross-validation makes use of all data which is a common practice especially when the data set is small and it is considered a robust assessment process.

2.4 Data Collection

According to USPTO, each patent is categorized into different technology classes. We select classes which are good representations of a technology-related industry. We started with classes including keywords "Wireless Network," "Advertising" and "Telecommunication." [3] The list of selected classes is provided in the Appendix A. Querying relevant patents for these classes yields a significant number of patents. Because of limitations in acquiring large amount of litigation information, we were not able to obtain all relevant litigation documents for all these patents. For this reason, we selected a random subset of these patents. We assume that the selected random subsets represent the interested population well. We specifically chose these keywords because of their relatively high rate of litigation and many patents.

Textual features and the number of inventors, number of claims, and referencing features are gathered directly from a patent. Financial data is collected from the SEC website which requires public companies to file periodic reports (i.e. quarterly and annual reports). The three features revenue, earnings per share and market share price capturing the financial situation of a company were extracted from annual financial reports (10-K for US companies and 20-F for foreign companies).

[3] We selected 25, 53, and 35 classes corresponding to 0.6, over 1, and 0.9 million patents for "Wireless Network," "Advertising," and "Telecommunication," respectively.

As this data is very limited due to incomplete information of private or small companies who are not obligated to report to SEC, we fit the models with three data options differing in the SEC data features: the model without the SEC data, with the SEC data, and with the SEC data after assuming a default value for missing cases. In the model with the SEC data, records without the SEC data are omitted before training the model. The default value is assumed to be a reasonable value in practice for each keyword separately. Finally, we collect litigation data from LexMachina[4] to label each instance in our data set. Table 2.2 illustrates the total number of samples of collected patents, the counts as well as the litigation rate associated with each keyword. Descriptive statistics detailing the number of litigated patents for each class in the time-to-litigation model is shown in Table 2.3.

Table 2.2 Summary of the data set

	Without SEC data				With SEC data			
Keywords	Litigated cases	Non-litigated cases	Total cases	Litigation rate	Litigated cases	Non-litigated cases	Total cases	Litigation rate
Wireless Network	509	26,154	26,663	1.91%	156	7,511	7,667	2.03%
Advertising	759	19,833	20,592	3.69%	134	6,068	6,202	2.16%
Telecommunication	646	51,654	52,300	1.24%	148	9,093	9,241	1.60%

Table 2.3 Summary of the data set for the time-to-litigation model

	Without SEC data				With SEC data			
Keywords	T<1 year	T<4 years	T<7 years	T<14 years	T<1 year	T<4 years	T<7 years	T<14 years
Wireless Network	120	206	80	103	31	53	30	41
Advertising	204	265	119	171	33	50	17	34
Telecommunication	155	238	95	158	18	35	30	65

[4] Lex Machina. https://lexmachina.com/; Accessed in 2014

2.5 Results

We calibrated the hyper parameters in order to obtain the best performance. SMOTE, the sampling technique, requires two hyper parameters: how many extra minority-class cases are generated, and how many extra majority-class cases are selected for each generated minority case. In our model, 5 and 1 are chosen, respectively. For example, assuming that the original data contains 6,500 and 140 records of the majority and minority classes, respectively, the number of minority class records is adjusted to be 840 (140 original cases plus 140*5=700) while the number of majority class records is adjusted to 7,200 (6,500 original plus 700*1=700). For the SVM classification model, a radial kernel with gamma of 0.001 and regularization value of 0.1 are selected. Weighting 0.3 for SVM and 0.7 for random forest gives the best ensemble model. The cut-off probability of 0.3 is chosen. For clustering part, hyper parameter X, A, and B defined in Section 2.3.2 are set to be 3.5, 1.3, and 0.015, respectively. This set of hyper parameters are applied to both the litigation and the time-to-litigation models.

Information gain is implemented for selecting the most significant features. Features related to reference knowledge are influential factors to the model to indicate the litigation likelihood and time to litigation including the first and second layer of references as well as the litigated references. With SEC data included in the model, features containing financial information also indicate high information gain. The significant non-textual features with their corresponding information gain value for the “Wireless Network” keyword are listed in Table 2.4. Note that the higher the information gain, the more significant the feature is.

Table 2.4 Features with their corresponding information gain

Features	Information gain
Revenue	0.0052
Earnings per share	0.0035
Litigated backward references	0.0032
Backward references	0.0027
2nd layer of backward references	0.0019

Figure 2.5 illustrates the top 30 textual features for the "Wireless Network" keyword. Unsurprisingly, common words generally related to the technology industry including "device," "system," "network," "server," "monitoring," "interface," "wireless" are contained in the top 30 features. Moreover, various words occurring in this list are relevant to communication systems, which are closely related to the "Wireless Network" keyword. These specific words include "video," "telephone," "remote," "audio" and they occur more often in litigated than in non-litigated cases. The proportion of litigated cases containing the specific words related to "Wireless Network" and the proportion of litigated cases containing the common words as defined in the beginning of the paragraph relevant to the technology industry are approximately 75 and 30 percent higher than that of non-litigated cases, respectively.

"device"	"claim"	"video"	"communication"	"comprising step"
"internet"	"method"	"monitoring"	"wherein"	"device method"
"network"	"via"	"system"	"associated"	"system claim"
"audio"	"remote"	"telephone"	"wherein user"	"method comprising"
"server"	"user"	"information"	"device system"	"system claim wherein"
"interface"	"wireless"	"personal"	"claim wherein"	"claim comprising step"

Figure 2.5 Top 30 features for "Wireless Network" keyword

The main goal for the litigation model is to differentiate between the litigated and non-litigated patents. The confusion matrix shows the performance of the model by comparing between the actual class (how a patent is originally labeled) and the predicted class (how the patent is predicted by the model). Positive and negative labels imply the litigation and non-litigation cases, respectively. We are interested in the probability that a patent is actually litigated when the predicted outcome is litigated, which is the precision. We also need to pay attention to the recall, which is the proportion of the patents that are predicted to be litigated among all litigated patents. Generally, there is an inverse relationship between these two values. Decreasing one value is compensated by a higher value of the other.

As it is not trivial to make a descriptive conclusion from considering both precision and recall, the F1-score representing the trade-off between the precision and recall is commonly used to compare the performance among different models. Different parameters such as the cut-off probability to define the predicted class and the regularization parameters as well as the hyper parameters used in the clustering approach are experimented to tune the model. The best values have been listed at the beginning of this section. For the litigation model, a comparison of the F1-score among different keywords and data options is depicted in Figure 2.6 while Figure 2.7 compares the precision and recall. Figure 2.7 illustrates no distinct pattern among data options, except the obvious inverse relationship between precision and recall. The data option of SEC without default gives the highest F1-score, which is consistent across the three keywords. Comparing among keywords, "Wireless Network" yields the best performance, followed by "Telecommunication" and "Advertising," respectively.

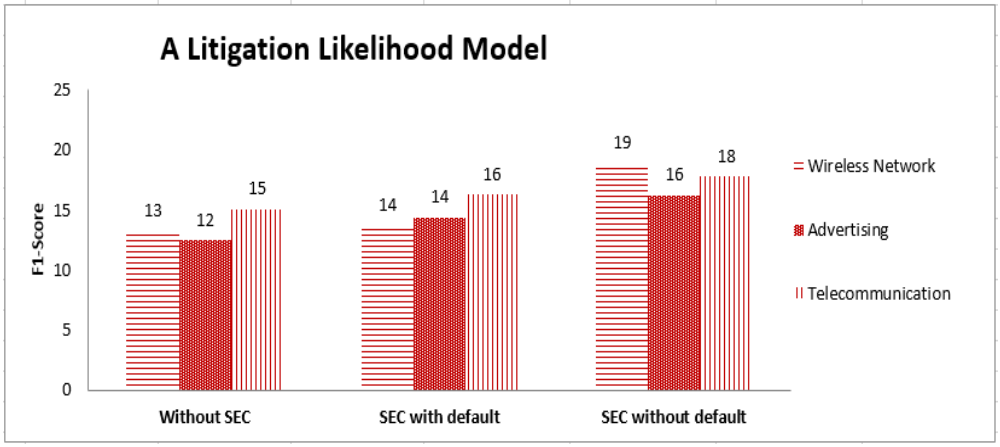

Figure 2.6 A comparison of F1-score as a percentage

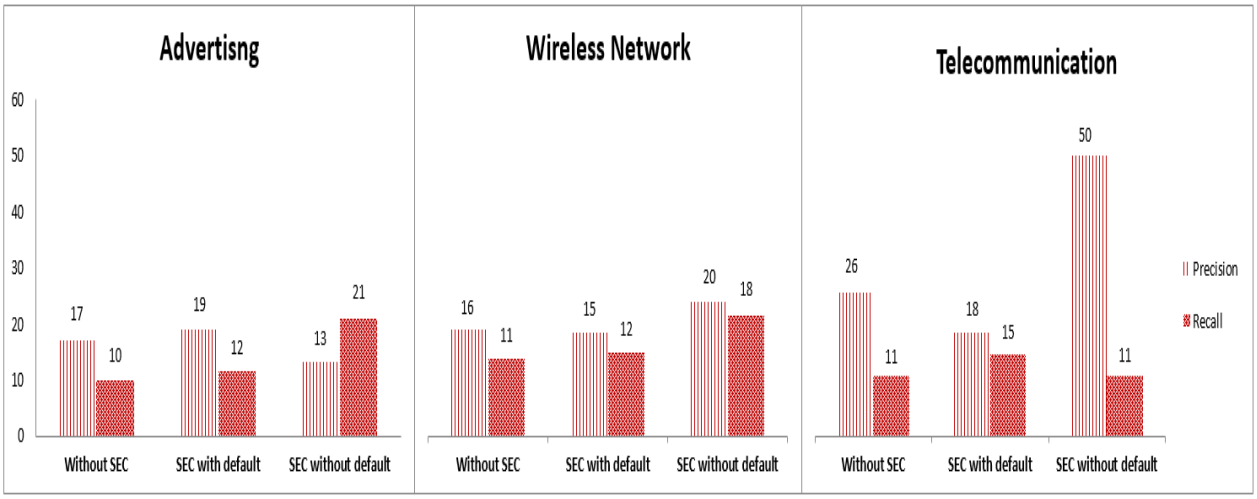

Figure 2.7 A comparison of precision and recall as a percentage

A comparison of F1-score between the cluster with ensemble method and the pure classification approach for SEC data without default across all 3 keywords is depicted in Table 2.5. The cluster with ensemble method for SEC without default data option outperforms the pure classification approach for “Wireless Network” and “Telecommunication” keywords. Figure 2.8 illustrates confusion matrices for “Wireless Network” with SEC without default option which perform best among all models. We observe a trade-off among 4 performance measures: true

positive, false positive, false negative and true negative values with different sets of hyper parameters.

Table 2.5 A comparison of F1-score between the cluster with ensemble and the pure classification approach across all keywords

	Pure classification	**Cluster with ensemble**
Wireless Network	0.1554	**0.1886**
Advertising	**0.1716**	0.1623
Telecommunication	0.1711	**0.1778**

		Actual	
		positive	**negative**
Predicted	**positive**	24 28	92 113
	negative	132 128	7419 7398

Figure 2.8 A confusion matrix for "Wireless Network" keyword and SEC data without default with different hyper parameters for the cluster with ensemble method

In the time-to-litigation model, the F1-scores obtained from the pure classification models are compared with those from the cluster with ensemble models for each time period as illustrated in Table 2.6. "Wireless Network" generally provides the best performance compared to the other keywords in almost all models and all data options except T < 1 class. The cluster with ensemble method performs better for the "Telecommunication" keyword while the pure classification method gives a better performance for the "Wireless Network" and "Advertising" keyword.

Table 2.6 A comparison of F1-score between two methods for time-to-litigation

T < 7 years class						
Keywords	**Without SEC**		**SEC with default**		**SEC without default**	
	pure classification	**cluster with ensemble**	**pure classification**	**cluster with ensemble**	**pure classification**	**cluster with ensemble**
Wireless Network	0.8807	0.8535	**0.8827**	0.8339	0.8609	0.8078
Advertising	**0.8629**	0.8263	0.8611	0.8279	0.8604	0.8479
Telecommunication	0.8475	**0.8492**	0.8484	0.8448	0.7301	0.7182
T < 4 years class						
Keywords	**Without SEC data**		**SEC with default**		**SEC without default**	
	pure classification	**cluster with ensemble**	**pure classification**	**cluster with ensemble**	**pure classification**	**cluster with ensemble**
Wireless Network	0.7708	0.7698	**0.7737**	0.7619	0.7269	0.6820
Advertising	**0.7548**	0.7454	0.7441	0.7439	0.7444	0.7511
Telecommunication	0.7382	0.7385	0.7442	**0.7539**	0.5098	0.5348
T < 1 year class						
Keywords	**Without SEC**		**SEC with default**		**SEC without default**	
	pure classification	**cluster with ensemble**	**pure classification**	**cluster with ensemble**	**pure classification**	**cluster with ensemble**
Wireless Network	0.3773	0.3793	0.3758	**0.3803**	0.3428	0.3373
Advertising	**0.4305**	0.4072	0.4165	0.4068	0.3691	0.3976
Telecommunication	0.3271	**0.3681**	0.3426	0.3623	0.1824	0.1630

Utilizing SEC data without default generally performs worse than other data options regardless of the choice of keywords. However, there is no obvious conclusion with respect to cluster with ensemble vs pure classification.

2.6 Discussions

In this paper, we develop a combined clustering and classification model to predict whether a patent is likely to be litigated and when it would happen. The best litigation model produces a 0.19 F1-score that is obtained from the "Wireless Network" keyword with SEC without default. Excluding SEC data gives worse performance than the other two models. This implies that financial information is beneficial.

Compared to Chien's model, which is the closest related work to ours, we use the data set that truly reflects the actual litigation rate. Not only significant imbalanced data causes the problem to be difficult, but also collecting a large number of correct data records is challenging. A litigation is found to be a very rare event with 1 to 2 percent. Our models with SEC without default yield approximately 0.13, 0.2 and 0.5 precision for three keywords (the probability of accurately predicting the litigation of a patent) under an acceptable value of recall. The improvement can be obviously recognized. With this value of precision the model incurs a relatively large number of false positives which is reflected by the recall value. The hyper parameters can be adjusted so that the desired balance level of precision and recall is achieved. This balance level depends mainly on the users' preference. For example, the users should pay more attention to increasing the precision value if their priority is to minimize the cost corresponding to missing the litigated patents.

An ability to correctly predict time to litigation of a patent by using the time-to-litigation model helps the users to save substantial resources. The model with SEC without default generally yields the worst performance regardless of keywords due to very limited data. Intuitively, the model corresponding to the top node gives a better F1-score than a bottom node. Among different classes, a high F1-score at a bottom node (node 6 in Figure 2.3) implies that the model performs well. Even though it is clear from the litigation model that enhancing the clustering with ensemble approach gives a better performance, this trend does not continue with the time-to-litigation model. We observe that the data in the time-to-litigation model is no longer strongly imbalanced and relatively small. This observation implies that the cluster with ensemble approach adapted from anomaly detection does not improve the performance in the time-to-litigate settings across the board.

The features related to the reference knowledge are important indicators for differentiating between litigated and non-litigated patents. Large numbers of references as well as a large number of litigated referenced patents imply a higher interest in that particular patent. The SEC data of each patent's assignee provides insight into the financial situation of the company owning the patent and improves the predictive power of the models.

2.7 Conclusions and Future work

The proposed litigation and time-to-litigation models attempt to predict the litigation likelihood and when it would occur. The clustering with ensemble approach are implemented in order to provide reliable predictive models. The problem is very challenging due to the low rate of litigation as well as the difficulty in obtaining a complete data set. Hence, better models can possibly be

achieved if more complete data sets are accessible. Future work can be done to improve the time-to-litigation model by considering a multi-class multi-label classification which takes hierarchical constraints into account and fit the global model [16]. An overall loss function is used to compare different models of each class separately. The loss function for a record , defined in [16] is a combination of penalty costs for misclassifying each node in the hierarchy tree depicted in Figure 2.3. The cost occurs at each class if the model misclassifies that particular class while its upper-classes in the hierarchy tree are correctly predicted. Another direction of future research includes predicting the litigation likelihood of a project based on contractual documents. In particular, textual as well as other relevant non-textual features can be extracted from the contracts establishing business deals and agreements among parties.

Chapter 3

A Semi-Supervised Learning Approach to Enhance Community-based Question Answering

3.1 Motivation and Related literature

A study by Pew Internet Project's research reported 87% of U.S. adults use the Internet, and 72% of Internet users sought health information online in the past year [17]. Other studies have also analyzed the modes in which health information is shared and its impact on consumer decision making [18, 19]. While it is known that patients are seeking information that might not be obtained during the course of their regular clinical care and valuable knowledge is publicly available online, it is not trivial for users to quickly find an accurate answer to specific questions. Consequently, Community-based Question Answering (CQA) sites such as Yahoo! Answers tend to be a potential solution to this challenge. In CQA sites, users post a question and expect the online health community to promptly provide desirable answers. Despite a high volume of users' participation, a considerable number of questions are left unanswered and at the same time other questions that address the same information need are answered elsewhere. This common situation drew our attention to develop an automated system for answering both unsuccessfully answered and newly posted questions.

Substantial research exists for developing systems that address physicians' information needs at the point of care [20-46]. Athenikos et al. [47] conducted a thorough survey reviewing

state of the art in biomedical question answering systems. Mishra et al. [34] systematically reviewed the ongoing in text summarization in biomedical domain. A majority of the studies used full-text articles (56%) and almost all of them used biomedical literature (91%) as input for summarization. However, limited research has been done in addressing the information needs of patients through automated approaches that synthesize the information shared across online health communities.

Outside of the health care domain, QA systems are widely studied in both open and other restricted domains. One of the common approaches is to retrieve answers based on past QA, which is also fundamental to our work. Shtok et al. [48] extracted an answer from resolved QA pairs obtained from Yahoo! Answers. Specifically, a statistical model was implemented to estimate the probability that the best answer from the past posts can satisfactorily answer a newly posted question. In addition to Shtok et al., Marom et al. [49] implemented a predictive model involving a decision graph to generate help-desk responses from historical email dialogues between users and help-desk operators. Feng et al. [50] constructed a system aiming to provide accurate responses to students' discussion board questions. An important element in these QA systems is identifying the closest (the most similar) matching between a new question and other questions in a corpus. However, this is not a trivial task since both the syntactic and semantic structure of sentences should be considered in order to achieve an accurate matching. A syntactic tree matching approach was proposed to tackle this problem in CQA [51]. Jeon et al. [52] developed a translation-based retrieval model exploiting word relationships to determine similar questions in QA archives. Various string similarity measures were also implemented to directly compute the distance

between two different strings [53]. A topic clustering approach was introduced to find similar questions among QA pairs [54].

An important component in QA systems is re-ranking of candidates in order to identify the best answer. A probabilistic answer selection framework was used to estimate the probability of an answer candidate being correct [55]. Alternatively, supervised learning-based approaches including support vector machine [56, 57] and logistic regression [58] are applicable to select (rank) answers. Commonly, collecting a large number of labeled data can be very expensive or even impossible in practice. Wu et al. [59] developed a novel unsupervised support vector machine classifier to overcome this problem. Other studies used different classifiers with multiple features for similar problems [60-64].

Luo et al. [65] developed an algorithm, SimQ, to extract similar consumer health questions based on both syntactic and semantic analysis. Vector-based distance measures were used to compute similarity score among questions. Statistical syntactic parsing and standardized unified medical language system (UMLS) were implemented to construct syntactic and semantic features, respectively. However, to effectively use the information in CQAs, we need to not only retrieve similar questions, but also provide and validate potential answers. SimQ was designed to retrieve similar questions from the NetWellness [66], a health information platform that has been maintained by clinician peer-reviewers. Questions collected within NetWellness tend to be clean and well structured, while CQA websites tend to be noisy.

Wong et al. has also contributed to automatically answering health-related questions based on previously solved QA pairs [67]. They provide an interactive system where the input questions are precise and short as opposed to accepting CQA questions directly as input.

In comparison to these systems, our work relies on implementing semi-supervised learning with Expectation Maximization (EM) approach [68]. Semi-supervised learning uses both labeled and unlabeled data for training. Given labeled and unlabeled data, EM based semi-supervised learning first trains an initial model using just the labeled set. This model is then used to estimate the label of each element in the unlabeled set. Next, the model is re-trained using both labeled and unlabeled set with the estimated labels from the previous step. The new model is used to refine the estimated labels in the unlabeled set. These steps are iteratively repeated until the algorithm converges or reaches pre-defined number of iterations. In addition, we employed Dynamic Time Warping [69] along with the vector-space distance [70] to measure similarity and incorporated biomedical concepts as additional features.

In summary, our work aims to automatically answer health-related questions based on past QA. We extracted candidate questions based on similarity measure and selected possible answers by using a semi-supervised learning algorithm. Automatically retrieving answers for questions from online health communities should provide the users a potential source of health information.

3.2 Methodology

We propose an algorithm to automatically generate answer for a health-related question based on past QA pairs. Detailed explanations for each step in the algorithm are extensively provided. We also give examples of annotations in the corpus used to train the model. Evaluation metrics are further discussed at the end of this section.

3.2.1 Algorithm

The system is built as a pipeline that involves two phases. The first phase implemented as a rule-based system, consists of: A) Question Extracting, which maps the Yahoo! Answers dataset to a data structure that includes question category, the short version of the question and the two best answers; B) Answer Extracting, which employs similarity measures to find answers for a question from existing QA pairs. In the second phase of Answer Re-ranking, we implement supervised and semi-supervised learning models that refine the output of the first phase by screening out invalid answers and ranking the remaining valid answers.

Figure 3.1 depicts the system architecture and flow. In training, phase I is applied for each prospective question in the training data set (with all other questions under a consideration corresponding to all questions in the corpus being different from the current prospective question). For test, the prospective question is a test question and all other questions are those from the training set. In this case, phase II uses the trained model to rank the candidate answer.

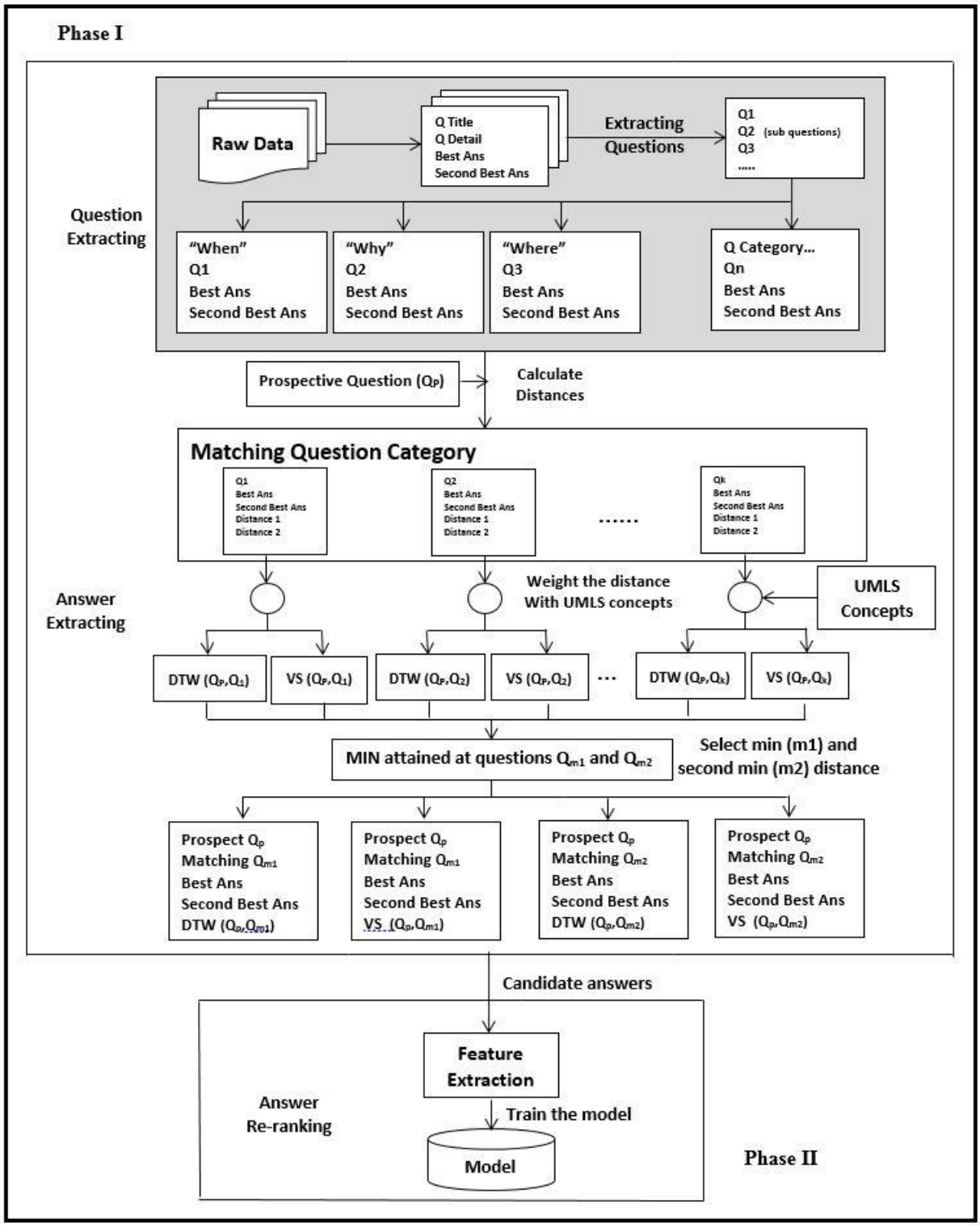

Figure 3.1: Overall architecture for training the system

We first describe the training phase. The Rule-based Answer Extraction phase (phase I) is split into 2 steps.

A) Question Extracting: For this system, we assume that each question posted on CQA sites has a question title and its description. Once users provide possible answers to the posted question, these responses are assumed to be marked as the best answer either by the question provider or community users. The second and subsequent best answers re chosen among remaining answers based on the number of likes. The raw data collected from CQA sites is unstructured and contains unnecessary text. It is essential to retrieve short and precise questions embedded in the original question title and its description (which can include up to 4-5 question sentences). Instead of using the whole question title and description which are long and verbose, we implement a rule-based approach to capture these possible short question sentences (sub-questions). These sub-questions are categorized into different groups based on question words such as "when" and "why." More specifically, regular expressions based on question words are used to classify sub-questions, which yield twelve main question classes. We consider sub-questions, instead of full questions and descriptions, for the rest of the paper.

B) Answer Extracting: Given a question, it is divided into sub-questions and matched with the question group using the above rule-based approach. Then, we compute the semantic distance between the prospective question and all other questions from the training sets belonging to the same group. Two distance approaches are employed in our work.

1. DTW-based approach: It is based on a sequence alignment algorithm known as Dynamic Time Warping [69], which employs efficient dynamic programming to calculate a distance between two temporal sequences. This allows us to effectively encode the word order without

adversely penalizing for missing words (such as in a relative clause). Applying it in our context, a sentence is considered as a sequence of words where the distance between each word is computed by the Levenshtein distance at a character level [71, 72]. For any two sequences defined as $Seq_1 = < w_1^1, w_2^1, \dots, w_m^1 > and\ Seq_2 = < w_1^2, w_2^2, \dots, w_n^2 >$ where m and n are the lengths of the sequences, Liu et al [69] defined the distance between two sequences (in our case, two sentences) as below:

$$D_{seq}(seq_1, seq_2) = f(m,n) \quad and \quad f(i,j) = d\left(w_i^1, w_j^2\right) + min\begin{cases} f(i-1,j) \\ f(i,j-1) \\ f(i-1,j-1) \end{cases}$$

where $f(0,0) = 0, f(i,0) = f(0,j) = \infty, i \in (0,m), j \in (0,n)$

Here $d\left(w_i^1, w_j^2\right)$ is the distance between two words computed by the Levenshtein measure.

2. Vector-space based approach: An alternative paradigm is to consider the sentences as a bag of words, represent them as points in a multi-dimensional space of individual words and then calculate the distance between them. We implement a unigram model with tf-idf weights based on the prospective question and other questions in the same category and compute the Euclidean distance measure.

We further take into account the cases that share similar medical information by multiplying the distances with a given parameter. If at least one word in the UMLS concepts of specific semantic types such as “organic chemical” and “pharmacologic substance” occurs in both the prospective question and a training question, we reduce the distance to account for the

additional semantic similarity. These UMLS concepts are specifically selected as we want to provide more weight to answers that mention a treatment approach.

The QA pairs in the training set corresponding to the smallest and the second smallest distance are extracted. Thus, we finally obtain a list of candidate answers, i.e. the answers referring to smallest and second smallest questions, for each prospective question. These answers are used as the output of the baseline rule-based system. This is repeated for each question in the training set, i.e. the prospective question corresponds to each question in the training set. At the end of this phase we have triplets (Q_p, Q_t, A_t) over all questions Q_p. Note that A_t is an answer to question Q_t with $Q_t \neq Q_p$ and each Q_p yields several such triplets.

The machine learning phase of answer re-ranking (phase II) is described next. The goal of this phase is to rank candidate answers from the previous step and select the best answer among them. Each triple (Q_p, Q_t, A_t) is aimed to be assigned as "valid" if A_t is a valid answer to Q_p, or "invalid" otherwise. We first select a small random subset of triplets and label them manually (there are too many to label all of them in this way). Both supervised and semi-supervised learning Expectation Maximization (EM) models are developed to predict the answerability of newly posted question as well as rank candidate answers. According to the semi-supervised learning model, we first train the supervised learning algorithms including Support Vector Machine (SVM) and Neural Networks (NNET) [73] based on manually labeling outputs from the above rule-based answer extraction phase. The trained model is used to classify the unlabeled part of the outputs of the phase I. Then, the classifier is re-trained based on the original labeled data and a randomly selected subset of unlabeled data using the estimated label from the previous iteration. These steps

are iteratively repeated in order to achieve a final estimated label. The supervised approach, on the other hand, only runs a classifier on the labeled subset and finishes.

A 10-fold cross validation is implemented in both semi-supervised and supervised approaches. Specifically, all labeled observations are partitioned into 10 parts where one part is set aside as a test set. The model is fitted based on the remaining 9 parts of the labeled observations (plus the entire unlabeled part for the semi-supervised learning approach). The parameters of the semi-supervised model are obtained by using the EM algorithm previously described. The fitted model is then used to predict the response in the part that we set aside as the test set. These steps are repeated by selecting different part to set aside as the test set. The features used in the models are illustrated based on the example below as shown in Table 3.1.

Example of a triple (Q_p, Q_t, A_t)

Prospective question: anxiety medication for drug/alcohol addiction?

Training question: Is chlordiazepoxide/librium a good medication for alcohol withdrawal and the associated anxiety?

Training answer: chlordiazepoxide has been the standard drug used for rapid alcohol detox for decades and has stood the test of time. the key word is rapid the drug should really only be given for around a week. starting at 100 mg on day one and reducing the dose every day to reach zero on day 8. in my experience it deals well with both the physical and mental symptoms of withdrawal. looking ahead he will still need an alternative management for his anxiety to replace the alcohol. therapy may help, possibly in a group setting

Table 3.1: A list of features used in the model

Type of Features	Features	Value
General Features	1.Text length of Q_p	5
	2.Text length of Q_t	12
	3. Number of stop words contained in Q_p	1
	4. Number of stop words contained in Q_t	5
	5. VS(Q_p, Q_t)	3.7052
	6. The difference between VS(Q_p, A_t) and VS(Q_t, A_t)	0.4303
	7. DTW(Q_p, Q_t)	29
	8. The difference between DTW(Q_p, A_t) and DTW(Q_t, A_t)	14.5
UMLS-based Features	9. Number of overlapping words in S_P and S_T	3
	10. Number of overlapping words in S_P and S_A	3
	11. Binary variable indicating whether a set of overlapping words in (S_P, S_T) and (S_P, S_A) are different	0
	12. Set difference of S_P and S_T	4
	13. Set difference of S_P and S_A	5

Sets S_P, S_T and S_A are sets of term corresponding to UMLS concepts occurred in Q_p, Q_t and A_t, respectively. Features 9 and 10 are calculated by counting the number of words contained in both sets. In order to obtain features 12 and 13, we find the elements that are in only one of the two sets.

The procedure to identify an answer to a newly posted question is illustrated in Figure 3.2 after the usual split of the corpus in train and test.

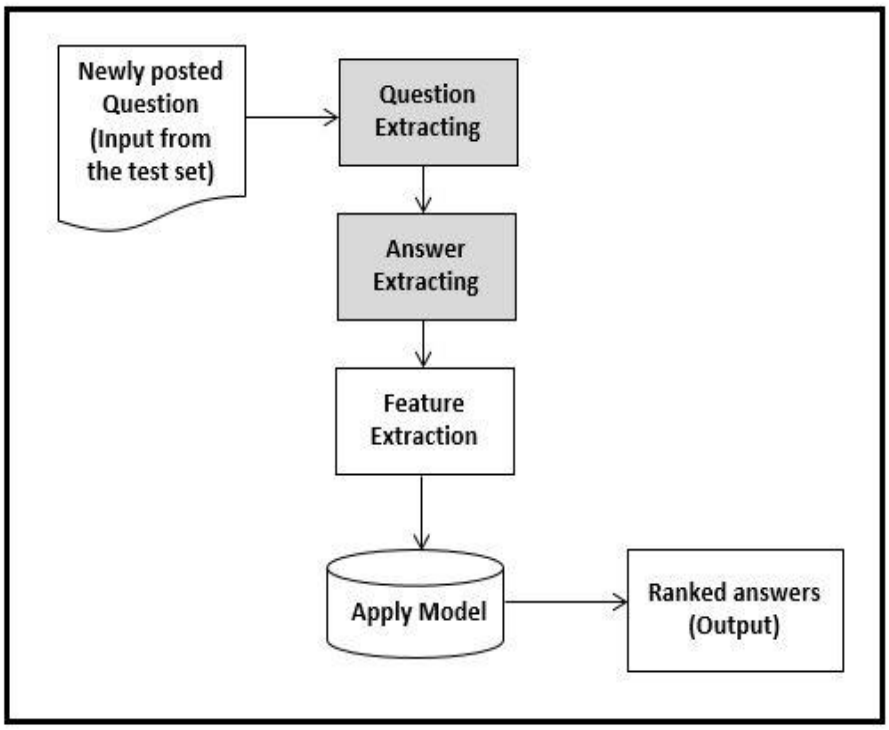

Figure 3.2: Process flow of the testing step

3.2.2 Corpus annotations

Table 3.2 depicts examples of annotations in the corpus. The inter-rater agreement for random instances (10% of total) assigned to two independent reviewers is very good (95% confidence interval of kappa from 0.69 to 0.93).

Table 3.2: Corpus annotation examples

A target question	A training question	A training answer	Label
can fully recovered alcoholics drink again	can a recovered alcoholic drink again?	what they say at aa is that there is no such thing as permanent recovery from alcoholism. there are alcoholics who never drink again, but never alcoholics who stop being alcoholics…..	valid
can fully recovered alcoholics drink again	if both my parents are recovered alcoholics, will i have a problem with alcohol?	yes, there is a good chance that you could inherit a tendency towards alcoholism….	invalid
anxiety medication for drug/alcohol addiction?	Is chlordiazepoxide /librium a good medication for alcohol withdrawal and the associated anxiety?	chlordiazepoxide has been the standard drug used for rapid alcohol detox for decades and has stood the test of time….	valid
anxiety medication for drug/alcohol addiction?	negative affects of alcohol and adhd medication?	drinking in moderation is wise for everyone, but it is imperative for adults with adhd…	invalid

3.2.3 Evaluation metrics

The following evaluation metrics is used to test the overall performance of our algorithm.

1. Question-based evaluation metrics

- For this paper, we define “Overall Accuracy” as ratio of the number of questions with at least one “correct” answer divided by total number of questions in the test set. A test question is labeled as “correct” if our algorithm predicts at least one valid triple correctly. For the case that

there is no valid answer in the question from the gold standard, we label it as "correct" if our algorithm predicts all corresponding triplets as invalid.

- The Mean Reciprocal Rank (MRR) with a set of test questions Q is defined as

$$\mathbf{MRR} = \frac{\mathbf{1}}{\mathbf{|Q|}} \sum_{\mathbf{i=1}}^{\mathbf{|Q|}} \frac{\mathbf{1}}{\mathbf{rank_i}}$$

where rank_i is the position of a valid instance in manually sorted probabilities from the model. If there are more than one valid instance in any question, minimum value of rank_i is used.

2. Triple-based evaluation metrics

Precision, recall, and the F1-Score can be used as standard measures for binary classification. We do not measure accuracy and ROC curves since the data set is heavily imbalanced.

3.3 Results

To test the algorithm, we obtain a total of 4,216 alcoholism-related QA threads from Yahoo! Answers. The sample outputs from our algorithm are shown in Figure 3.3, which indicates how our system could potentially be used by online advice seekers. In order to extract initial candidate answers in the rule-based answer extraction, our algorithm returns 8 instances for each prospective question (obtained from 2 different similarity measures where we extract at least 2 closest questions for each measure with 2 answers for each question). An example of output reported from the rule-based answer extraction is depicted in Figure 3.4.

Input (a newly posted question)	Output (a possible answer)
Question title: Is there anything for alcohol withdrawal, medication recovery **Question description:** Just like how there is some medication for drug addicts, like Suboxone, Methadone... etc. I wonder if theres anything like that for alcoholic? ..weed? lol.. or is there? - just wondering, thanks.	chlordiazepoxide has been the standard drug used for rapid alcohol detox for decades and has stood the test of time. the key word is rapid the drug should really only be given for around a week. starting at 100 mg on day one and reducing the dose every day to reach zero on day 8. in my experience it deals well with both the physical and mental symptoms of withdrawal. looking ahead he will still need an alternative management for his anxiety to replace the alcohol. therapy may help, possibly in a group setting.
Question title: Symtons off alcohol abuse **Question description:** what side affecs can i expect when in stop drinking	i hope your meaning long term? not if you drink one night then suddenly stop all these return to normal, because they reactions will be piss poor when your pissed. fitness: will be down, the same as when you eat unhealthy etc.. you need to exercise to improve this again to a good standard. liver: if you havent abused alcohol for long & your still young, your liver can return to health naturally.. if it has been abused for years, you may have some liver disease, where it cant recoved.. reflexes: might be slightly worse of.. but shouldnt be affected too much, after normal exercise etc, should be back to normal.
Question title: Alcohol effects baby **Question description:** So I'm four months preggers and evryones been telling me tht i shud not drink becuz its bad for the baby. i know pregnant women shudnt drink but thts just becuz they might do something they might not normally do thts bad, lik sleep with a random person or drive and crash. but wat if i drink and my friend watches over me and makes sure i don't do anything bad? Or shud i just not risk it?	no! if you suspect youre pregnant, dont go anywhere near alcohol -period. youre going to ruin the life of an innocent child who deserves more if you do.

Figure 3.3: System output

(1) Case where accurate answers are extracted

Question: is there anything for alcohol withdrawal, medication recovery

Matched question: is chlordiazepoxide/librium a good medication for alcohol withdrawal and the associated anxiety?

Best Answer: Chlordiazepoxide has been the standard drug used for rapid alcohol detox for decades and has stood the test of time…

2nd Best Answer: This is pretty standard treatment for acute withdrawal/anxiety, etc. Librium, a benzodiazepine, is commonly prescribed for this.

(2) Case where accurate answers failed to be extracted

Question: are there a lot of women who do not want to breastfeed because they want to drink alcohol?

Matched questions: do you look down on men who don't drink alcohol?

Best Answer: oh yeah, because every woman…

2nd Best Answer: I personally don't look down…

Figure 3.4: An example result returned from the algorithm to determine candidate answers

A randomly selected set of 220 threads are used as labeled questions. Overall, 119 out of 220 questions, or 54.1 percent, have valid answers among those extracted in the rule-based answer extraction phase. After retrieving candidate answers, we further aim to re-rank them and select the best answer (if there is a valid answer). Specifically, the semi-supervised learning model (EM) is trained on 1,553 labeled and 10,000 unlabeled triplets. In the training data, 297 triplets are

manually labeled as "valid" and 1256 as "invalid." The typical 10-fold cross validation is implemented in order to validate the model.

We analyze the feature set by using information gain to indicate a significance of each feature. The most influential features are the number of stop words contained in Q_p, the text length, the distance of (Q_p, Q_t), as well as the number of overlapping UMLS words between Q_p and Q_t, i.e. in S_P and S_T. All information gains for these significant features are listed in Table 3.3.

Table 3.3: Information gain score of 5 significant features

Features	Information gain
1. Number of stop words contained in Q_p	0.0912
2. Text length of Q_p	0.0804
3. DTW(Q_p , Q_t)	0.0395
4. Number of overlapping words in S_p, S_T	0.0393
5. VS(Q_p, Q_t)	0.035

The best model is selected by varying the cutoff probability of being valid or invalid to obtain the maximum F1-score. We select NNET and SVM approaches to train the model on a subset (these two models provide the best performance). For the SVM classifier, the probability is obtained by fitting a logistic distribution using maximum likelihood to the decision values provided by SVM.

The semi-supervised learning (EM) algorithm with 10 iterations trained with NNET gives the best performance. Overall, NNET performs better than SVM regardless of the model

implemented (semi-supervised vs pure classification). The model parameters tuned by the EM algorithm converge with more EM iterations, as listed in Table 3.4.

Table 3.4: Evaluation metrics

Evaluation Metrics	Supervised learning		Semi-supervised learning (EM)			
	NNET	SVM	1 iteration		10 iterations	
			NNET	SVM	NNET	SVM
Overall Accuracy	0.7168	0.6711	0.7161	0.7018	**0.7216**	0.7158
MRR	0.585	**0.6331**	0.5942	0.6139	0.6	0.6
F1-score	0.3858	0.38	0.4105	0.4049	**0.4163**	0.4064
Precision	0.3708	0.4645	0.3936	0.4156	0.4019	0.4134
Recall	0.4034	0.3215	0.4299	0.3946	0.4365	0.3995

NNET		Actual	
		positive	negative
Predicted	positive	128	191
	negative	169	1065

SVM		Actual	
		positive	negative
Predicted	positive	119	168
	negative	178	1068

Figure 3.5: Confusion matrices for 10 iterations of EM trained with NNET and SVM

We are also interested in understanding whether UMLS-based features (feature 9-13 listed in Table 3.1) play a role in predicting the validity of a candidate answer. Hence, we train another model, which excludes all UMLS-based features, and compare the results (obtained from 10 iterations of EM) with the original model as illustrated in Figure 3.6. With UMLS-based features, the model gives a better performance, which is consistent across all evaluation metrics. This implies that these features play a role in distinguishing between valid and invalid answers.

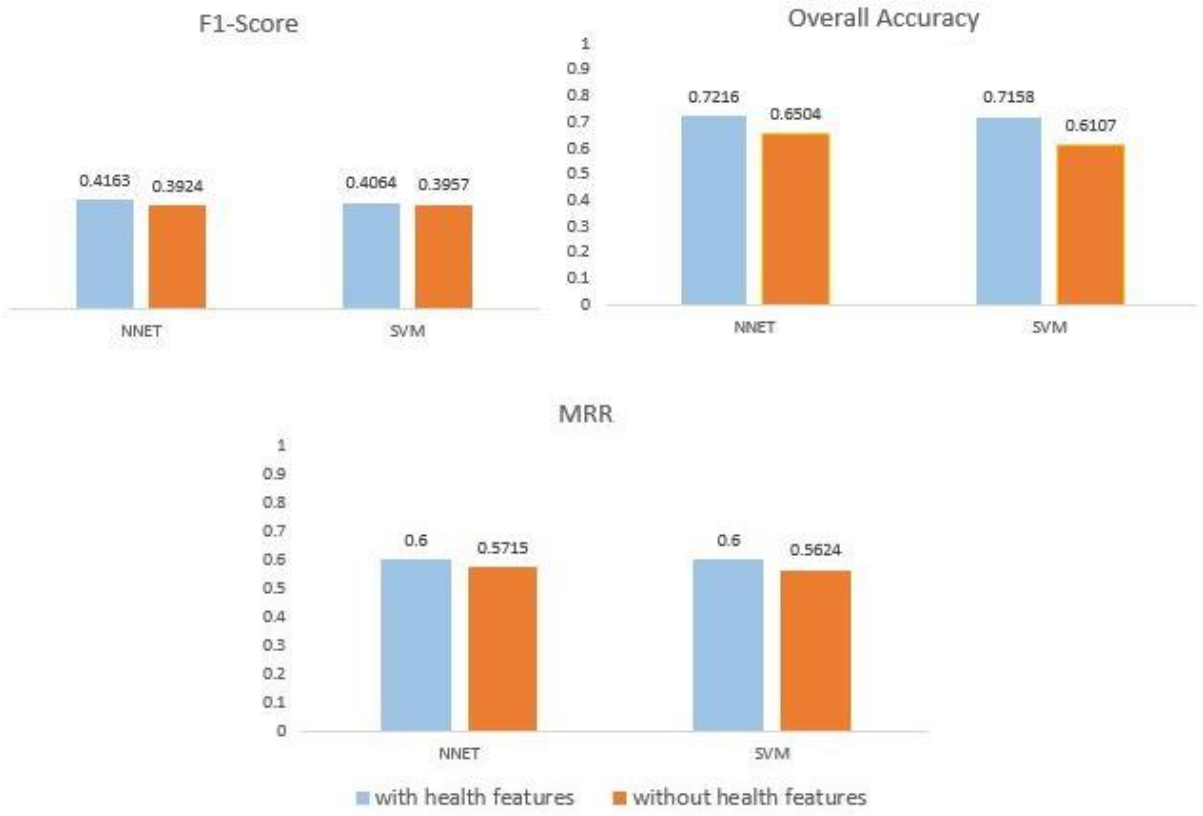

Figure 3.6: Performance between the original and adjusted model to test significance of UMLS-based features (health features)

3.4 Discussions

In this paper, we develop an automated QA system by using previously resolved QA pairs from a CQA site and evaluate it. Even though we use Yahoo! Answers as a data source, our algorithm can be adapted and applied to other CQA sites. Overall, the system achieves 72.2 percent accuracy and 0.42 F1-score, which are significant given that the problem is challenging and the data is imperfect. Internet users typically provide responses in an ill-formed fashion. Our data also consists of a significant number of complex questions, e.g. a user discusses about his or her situation in 10-20 sentences and then asks whether s/he is an alcoholic. Moreover, some questions are very detailed; for example, the percentage of alcohol resulting from a given combination of chemical components.

Comparing with Luo et al. [65] who retrieved the similar questions based on the distance measure, we rely on this idea with different approaches. In order to compute the similarity score between questions, we employ the DTW measure instead of relying on the vector-based distance measure. As our data is relatively long and noisy, we decide not to include the syntactic features proposed by Luo et al. in our model. While the accuracy reported by Luo et al. is relatively better than our system's, it might be mostly due to the noisy nature of online health community discussions.

Shtok et al. [48] used resolved QA pairs to reduce the rate of unanswered questions in Yahoo! Answers. The experiment in Shtok et al. was also tested with health related questions and the accuracy as measured by the F1-score was 0.32. Our method, which trains a semi-supervised learning model with a smaller amount of manually labeled data, results in 0.4 F1-score. A better performance might be because of several reasons. We first categorize questions in a corpus into

different groups based on question keywords. Instead of computing the distance between a test question and all other questions in the corpus, categorizing questions reduces the scope of questions an algorithm needs to search. As we categorize collected questions into different groups based on question keywords, latent topics and "wh" question matching features used in Shtok's are not valuable in our context. Our algorithm also uses multiple features related to the UMLS medical topics in order to enhance the model's performance when applied within the health domain. While Shtok et al. relied on cosine distance, the Euclidean distance performs better in our evaluation. Among distance measures used in our work, more valid answers can be correctly identified with the DTW-based approach than the vector similarity measure, which can be observed when manually annotating the output from the rule-based answer extraction. In addition, our algorithm extracts multiple candidate answers retrieved from two closest QA pairs for each distance metric and the two best answers for each question. In each QA pair, both the best and the second best answer are extracted compared to Shtok et al. where only one best answer was extracted. In the re-ranking phase, we implement semi-supervised learning to gain benefits from unlabeled data while Shtok et al. only relied on a supervised learning model.

Using a semi-supervised learning model that leverages unlabeled data is reasonable against other traditional supervised learning models because obtaining labeled data is very expensive and time-consuming in practice. Since the features of the machine-learning algorithm are not specific to alcoholism, our system should be applicable for other related topics. On the other hand, it would be possible to increase the accuracy for "alcoholism" if we use specific features such as concepts related to alcoholism.

For the machine-learning component, number of stop words contained in a test question is the best indicator for differentiating between valid and invalid answers. The distance between a test question and other questions in the training data set is also important in distinguishing valid and invalid answers. The closer the distance is, the higher the chance of the corresponding answer being valid. We opine that a short text length is likely to contain more precise information, which implies the significance of text length feature. Matching UMLS terms, which imply a closer similarity between questions, play a role in determining the validity of the answer. Even though UMLS-based features show lower information gain, the overall accuracy is improved by 7% and 10% with NNET and SVM when these features are included.

Limitations and future work

The main limitation of our work is the lack of assessment of the model's generalizability. Even though our algorithm is generic and does not include any features that are specific to the topic of alcoholism, we have not validated it in different domains. Approximately 30 percent (obtained from a preliminary observation) of all questions cannot be answered based on existing answers; some of these questions also require additional resources that are more technical and reliable, such as medical textbooks, journals and guidelines.

3.5 Conclusions

The question-answering system developed in this work achieves reasonably good performance in extracting and ranking answers to questions posted in CQA sites. Our work seems to be a

promising approach for automatically answering health-care domain questions asked by online healthcare communities. The system and the gold standard corpus are available in github [74].

Chapter 4
Truth Validation with Evidence

4.1 Introduction

Accessing information online is expanding tremendously in various domains as reported in a study by the Pew Internet Project's research [75]. According to the study, offline population in the U.S. has declined significantly since 2000 to the extent that in 2016 only 13% of U.S. adults did not use the internet. Internet usage gives people an opportunity to extensively seek information online; however, posted contents available on web pages are not necessarily reliable. As online information spreads rapidly, its quality is considerably crucial. Misinformation potentially leads to serious consequences significantly affecting internet users. The main motivation of our study is to validate the truthfulness of textual information obtained from various sources as well as to provide supporting evidence.

Humans can identify the truthfulness of a statement particularly for common fact cases. Nevertheless, manually inspecting statements is a time-consuming process that becomes impossible for large-scale data. Determining the truthfulness of each statement in an automated fashion is a promising alternative solution. This problem is highly challenging due to the lack of an encompassing and comprehensive corpora of all true statements. Despite of its challenges, it draws a lot of attention from prior studies to develop truthfulness-validating systems. The previously proposed systems mainly rely on web search engines to verify whether statements are

true or false. Additional information regarding sources which statements are extracted from is also taken into account in most algorithms.

In comparison to these truthfulness-validating systems, our work relies on knowledge from reliable sources rather than web search engines. Statements gathered from reliable sources have various length and may be verbose and thus we represent each of these statements as triplets consisting of a subject entity, an object entity, and their relation. These triplets capture the main contents embedded within statements. A Knowledge Graph (KG) is then constructed from these triplets where nodes are entities and arcs represent the relationships between nodes. In our algorithm, a relation extraction method is used to extract triplets from the statement we aim to verify the truthfulness of. We call this statement and its corresponding triplets as "a lay statement" and "lay triplets." The truthfulness of each lay triplet is verified based on an inference method corresponding to KG constructed from reliable sources.

After identifying the truthfulness of lay triplets, our algorithm additionally provides supporting evidence. Determining evidence for true triplets is relatively straightforward compared to identifying the evidence of falseness. Considering a true triplet, a supporting evidence is a set of paths between the subject and object entities inferred from KG associated with reliable sources. On the other hand, it is unclear how to obtain evidence for false triplets. Reasonable evidence for each false triplet should be a collection of relevant triplets extracted from KG under a specific condition. We explain our key idea with an example. Consider the false triplet ("property", "has_a", "space rocket"). We find in KG all triplets ("property", "has_a", $\bar{o}$). In this case, a set of all possible candidates $\bar{o}$ denoted as $\bar{O}$ can be {"bedroom," "kitchen," "bathroom," "roof," "garden," "shed," "swimming pool"}. A long proof of evidence can be this candidate evidence set

and the fact that "space rocket" $\notin \bar{O}$. The drawback here is that the size of $\bar{O}$ can be very large. Summarizing the candidate collection into a concise but meaningful evidence set is challenging especially when the size of the collection is large.

In order to overcome this difficulty, we develop a novel algorithm to extract supporting evidence from concepts in ontologies. For any false triplet, we rely on the idea of representing each candidate with its broader concepts in ontologies given that the false triplet concept is not part of these broader concepts. Considering our example, an ontology could provide us with the fact that the first four terms in $\bar{O}$ are related to "house" and the remaining terms correspond to "backyard." Finally, as an evidence we provide ("property", "has_a", "house") and ("property", "has_a", "backyard") and the facts "space rocket" is not "house," "space rocket" is not "backyard." Given the false triplet and its candidates, matching concepts in ontologies are considered. Then, we gather a set of potential evidence which includes candidate concepts and their broader concepts (satisfying some conditions). Evidence of various levels of granularity is constructed by a graph based algorithm on the subsumption tree of the ontology. We select an optimal collection of evidence from the potential evidence set. The optimized set of evidence is the smallest sub-collection of the potential evidence set under the assumption that all candidates $\bar{o}$ have to be covered by themselves or their broader concepts which leads to a set covering problem.

In the rest of the paper, we consider the following running example. Given a false triplet ("Google", "OfficeLocationInUS", "Minneapolis"), we generate the evidence of falseness based on its relevant triplets from KG. The relevant triplets retrieved from KG have the "OfficeLocationInUS" relation associated with the "Google" or "Minneapolis" entity. In particular, we first find locations of Google's offices such as "Atlanta," "Chicago," "Los Angeles,"

"Miami," "Mountain View," etc. Also, companies whose office is located in Minneapolis such as "Target Corporation," "U.S. Bancorp," "Xcel Energy" are considered. These retrieved entities are used as the falseness evidence as we claim that "Minneapolis" is not part of all retrieved locations and similarly "Google" is not part of the set of retrieved companies. We rely on knowledge from ontologies to generate a concise set of evidence. For example, an ontology about geography is used to state that "Google" has offices in many states across U.S. while "Minneapolis" is located in Minnesota which is not one of these states.

Our main contribution is to provide supporting evidence for a given lay triplet after its truthfulness has been identified. If the triplet is true, then paths in KG provide evidence, however if false, then it is much more challenging to come up with the concept of evidence. To the best of our knowledge, no prior work provided supporting evidence of any given false lay statement by taking into account KG and ontologies. Our proposed system which combines knowledge from ontologies with predicate triplets from a KG contributes in this space. We specifically focus on selecting a complete set of falseness evidence to be as concise as possible. Also, our system relies mainly on both KG and ontologies which are constructed from reliable sources instead of knowledge from unverified web pages.

Our algorithm to provide supporting evidence along with the truthfulness of the lay triplet is applicable in various domains such as politics, sciences, news, and health care. Our work focuses on the health care domain as a case study mainly because of abundant health-related information available online and the importance of information quality. Specifically, a large number of medically related web sites are easily accessible online but only half of these sites have content

reviewed by professionals [76]. In addition, distorted information related to health conditions potentially causes devastating effects.

We summarize the literature in Section 4.2. In Section 4.3, we describe relevant background information, problem definitions, and thoroughly discuss our main algorithm. Data preparation and results of the algorithm based on our case study are reported in Section 4.4 while further discussions are provided in Section 4.5. Conclusion and future work are stated in Section 4.6.

4.2 Related Work

Our algorithm verifies the truthfulness of any lay statement based on a KG thus we survey prior work in truth discovery-related fields. Substantial work exists in truth discovery for determining the veracity of multi-source data. In particular, the truth discovery problem aims to identify whether assertions claimed by multiple sources are true or false. Reliability of sources is also determined. Waguih and Berti [77] provide an extensive review and an in-depth evaluation of 12 truth discovery algorithms. Additional truth discovery methods are proposed varying in many aspects to jointly estimate source reliability and truth statements [78-84]. These methods rely on a common assumption that information provided by a reliable source tends to be more trustworthy and the source providing trustworthy information is likely to be more reliable.

TruthOrRumor, a web-based truth judgment system, determines the truth based on results from a search engine [85]. It considers reliability of data sources based on historical records and the copying relationship. Also, it implements currency determination techniques to take into

account out-of-date statements. Wang et al. [86] propose an algorithm to determine the truthfulness of a given statement based on a combination of a support score and credibility ranking value. While the support score measures how a web search result supports the statement, the credibility ranking computes the reliability of web pages. The t-verifier system [87] requires users to pre-determine specific parts of statements to be verified. These systems take into account additional information of a data set or its source when determining the truthfulness of the statement.

Yin and Tan aim to distinguish true from false statements given a small set of ground truth facts [88]. A graph optimization method is used in [88] where each node in the graph represents a statement and each edge connects a pair of relevant statements. Statements in the set of ground truth facts are labeled as 1. The algorithm assigns a truthfulness score ranging from -1 to 1 to each unlabeled statement. The scores of unlabeled statements not directly related to any labeled statements are possibly close to 0. This implies that the truthfulness of these statements remains undefined. Yamamoto and Tanaka propose a system to determine the credibility of a lay statement and extract aspects necessary to verify the factual validity from web pages [89] whenever the statement is true. In order to estimate validity of a lay statement, the system collects comparative fact candidates using a web search engine. Fact candidates are sentences retrieved from the search engine that match a pattern specified by the lay statement. Then the validity of each candidate is computed based on the relation between the pattern and the entity contained in the candidate.

In comparison to the previous work, a focus of our algorithm is to provide concise but reliable supporting evidence in addition to identifying the truthfulness of a lay statement. The algorithm proposed by Yamamoto and Tanaka is similar to our system when the statement is true. In particular, both [89] and our work use comparative candidate facts in order to assess the

credibility of any lay statement. Instead of using web search engines, we rely on an inference method with respect to a KG to collect candidate triplets. A truthfulness score for the lay triplet is computed and compared against scores from those candidate triplets in order to determine whether the lay triplet is true or false. None of these works provide evidence of false statements which is the main contribution of our work.

4.3 Methodology

Content commonly found in textual documents especially online texts can be unreliable. In this study, we aim to identify whether a given lay statement is true or false and provide supporting evidence. We collect lay statements from many web pages publicly available online. We use a relation extraction algorithm [90] to extract triplets consisting of subject entity s, object entity o and their relation $r(s, o)$ embedded within lay statements. Our problem is scoped down to identifying the truthfulness of triplets representing the original lay statements. Knowledge obtained from reliable sources is an important factor in determining the trustworthiness of the lay triplets. We assure that the reliable resources are structured in a form of triplets (s, r, o) which are used to construct a knowledge graph. Nodes and edges in KG represent entities and their relations, respectively. We write $(s, r, o) \in$ KG to mean that s, o are nodes in KG and r corresponds to an edge between them.

An evidence of falseness is obtained based on knowledge from various ontologies in related domains. In order to properly discuss falseness evidence and the main algorithm, we first provide a brief overview of a knowledge base (KB) or ontology and relevant background information.

According to terminological knowledge, elementary descriptions are concept names (atomic concepts) and role names (atomic roles). Concept descriptions are built from concept and role names with concept and role constructors. All concept names and concept descriptions are generally considered as concepts. A deeper knowledge of ontologies can be obtained from [91].

Let KB = $(\mathcal{T}, \mathcal{A})$ be a knowledge base with $\mathcal{T}$ being a TBox and $\mathcal{A}$ an ABox as defined in [91]. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a model of KB corresponding to an ontology. We assume that KB is consistent. We assume that for each $(s, r, o) \in$ KG there are concepts D, E in KB such that $s = D^{\mathcal{I}},\ o = E^{\mathcal{I}}$. We denote $s = D^{\mathcal{I}}$ if and only if $D = C(s)$, where D is a concept, i.e., given entity $s \in$ KG, $C(s)$ is the corresponding concept. We define special concepts $\top$ and $\bot$ as top (universal) and bottom (empty) concepts. Concept constructors such as an intersection $\sqcap$, a union $\sqcup$, and a negation $\neg$ combined with concept names are used to construct other concepts. Let V_C be the set of all concept names. We also define $a \not\sqsubseteq b$ for concepts a and b if and only if $\exists y, y \neq \bot$ where y is a concept such that $y \sqsubseteq a \sqcap \neg b$.

A KB classification algorithm computes a partial order $\leq$ on a set of concept names with respect to the subsumption relationship, that is, $A \leq B \Leftrightarrow A \sqsubseteq B$ (A sub-concept of B) for concept names A and B. The classification algorithm incrementally constructs a graph representation in a form of a direct acyclic graph, called the subsumption tree, of the partial order induced by KB [92]. Note that in this paper we use the term “tree” to use the term consistent with past literature. The underlying structure is actually an acyclic graph. Given X as a set of concepts, computing the representation of this order is equivalent to identifying the precedence relation $\prec$

on X, i.e., $a \prec b$ for $a, b \in X$ if and only if $a \sqsubseteq b$ and if there exists $z \in X$ such that $a \sqsubseteq z \sqsubseteq b$, then $z = a$ or $z = b$.

Given the precedence relation $\prec_i$ for $X_i \subseteq X$, the incremental method defined in [92] computes $\prec_{i+1}$ on $X_{i+1} = X_i \cup \{\mathrm{c}\}$ for some element $\mathrm{c} \in X \setminus X_i$. The method consists of two main parts which are a top and a bottom search. The top and the bottom search identify sets $X_i \downarrow c = \{x \in X_i |\ c \sqsubseteq x$ and $c \not\sqsubseteq y$ for all $y \prec_i x, y \in X_i\}$ and $X_i \uparrow c = \{x \in X_i |\ x \sqsubseteq c$ and $y \not\sqsubseteq c$ for all $x \prec_i y, y \in X_i\}$. At the i^{th} iteration, arcs corresponding to $\prec$ between c and each element in $X_i \downarrow c$ as well as c and each element in $X_i \uparrow c$ are added. Also, some existing arcs between elements in $X_i \downarrow c$ and $X_i \uparrow c$ are eliminated. At the end we have $a \prec b$ if and only if there is an arc in the constructed subsumption tree.

Our proposed system is built as a pipeline involving two main steps. We denote by (s, r, o) the lay statement triplet that requires evidence.

1. Determining the truthfulness of the triplets: We rely mainly on the inference method called the Path Ranking Algorithm (PRA) introduced by Lao et. al. [93] to verify whether each triplet $(\bar{s}, \bar{r}, \bar{o})$ in KG is true or false. The PRA produces $score_{PRA}$ for every pair of nodes. A PRA model is trained at each relation level. Particularly, the PRA model for a relation type $\bar{r}$ is trained to retrieve other nodes which potentially have a relation $\bar{r}(\bar{\mathrm{s}}, \cdot)$ given node $\bar{s}$. We retrieve $\tilde{o}$ related to $(\bar{\mathrm{s}}, \bar{r}, \tilde{o})$ with $score_{PRA}(\tilde{o}; \bar{\mathrm{s}}) \geq \varepsilon_1$. All such object candidates $\tilde{o}$ are denoted by $\bar{O} = \bar{o}(\bar{\mathrm{s}})$. A subject candidate set $\bar{S}$ is extracted in a similar way. The triplet (s, r, o) is labeled as "True" if $score_{PRA}(\mathrm{s}; \mathrm{o}) \geq \varepsilon_2$ or $score_{PRA}(\mathrm{o}; \mathrm{s}) \geq \varepsilon_2$, and "False" otherwise. In addition, paths

corresponding to high PRA scores are provided as supporting evidences of truthfulness if (s, r, o) is true.

2. Extracting the evidence of falseness: We now assume that (s, r, o) has been labeled as False in step 1, and either $s \notin \bar{S}$ for extracting a subject evidence of falseness or $o \notin \bar{O}$ for extracting an object evidence of falseness. Set $\bar{O}$ is the set of all objects that verify s and r. If (s, r, o) is false, then it has to be the case that $o \notin \bar{O}$ as otherwise (s, r, o) would be true. Same holds for $\bar{S}$. We only discuss in detail the object evidence while the subject evidence is defined similarly.

Our validation for false statements do not rely on PRA, i.e. any inference algorithm on KG can be used. We found PRA to work best on our data. We next formally define evidence for false triplets. Recall that $\bar{O}$ is the set of all objects $\bar{o}$ for which $(s, r, \bar{o}) \in$ KG. It is important that these are all. In essence as a proof of falseness we can provide $\bar{O}$ together with the fact o $\notin \bar{O}$. However, this in many cases would provide a very long evidence since $|\bar{O}|$ is typically large. Instead we want to "aggregate" $\bar{O}$ into some smaller set α and still claim that o $\notin \alpha$. Wrapping these intuitions in the ontology formalism yields the following definition.

Definition 4.1: An object evidence of falseness is a collection $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ of concept names in KB such that

1) for each $\bar{o} \in \bar{O}$ there exists $i \in \{1, \dots, k\}$ such that $C(\bar{o}) \sqsubseteq \alpha_i$,
2) there exists a concept $y, y \neq \bot$ such that $y \sqsubseteq C(o) \sqcap \neg(\coprod_i \alpha_i)$ for all $i = 1, \dots, k$.

The second condition can be rewritten as $C(o) \not\sqsubseteq \alpha_1 \sqcup \dots \sqcup \alpha_k$ which in turn is equivalent to $C(o) \not\sqsubseteq \alpha_1$ and…and $C(o) \not\sqsubseteq \alpha_k$. In words, the second condition is equivalent to the requirement

that $C(o)$ is not part of any element in the evidence set α. The collection α is considered as an aggregated set of $\bar{O}$. We further define "potential evidence" α_i if there exist object evidence of falseness α such that $\alpha_i \in \alpha$.

From the definition, $\bar{O}$ is a set of object candidates having a relation $r(\mathrm{s}, \bar{o})$ for the given triplet (s, r, o). The first requirement in Definition 4.1 assures that each candidate $C(\bar{o})$ has to be subsumed by at least one potential evidence α_i (α is an aggregation of all elements in $\bar{O}$). As an example, letting $\alpha = \bar{O}$ satisfies the first requirement as each $C(\bar{o})$ for $\bar{o} \in \bar{O}$ is always subsumed by itself. According to the second requirement, concept $C(o)$ is not subsumed by any potential evidence α_i. This ensures that concept $C(o)$ obtained from the false triplet (s, r, o) does not belong to the evidence collection α (it mimics o $\notin \alpha$). Among all collections α we want to find the smallest one which is formalized later.

Referring to the false triplet example ("Google", "OfficeLocationInUS", "Minneapolis"), we let $\bar{O}$ be all object candidates retrieved from KG which have the relation "OfficeLocationInUS" associated with subject "Google." Given $\bar{O} =$ {"Ann Arbor," "Atlanta," "Austin," "Birmingham," "Boulder," "Cambridge," "Chapel Hill," "Chicago," "Irvine," "Kirkland," "Los Angeles," "Miami," "Mountain View," "New York," "Pittsburgh," "Playa Vista," "Reston," "San Bruno," "San Francisco," "Seattle," "Sunnyvale," "Washington DC"}, selecting $\alpha = \bar{O}$ satisfies both requirements in Definition 4.1. The second requirement is satisfied as $C(o)$ associated with $o =$ "Minneapolis" is not subsumed by any element in the evidence collection α. Moreover, the collection {"West region," "Northeast region," "South region," "Michigan," "Illinois"} is an example of a smaller evidence set which satisfies both requirements.

We propose Algorithm 4.1 to extract the evidence of falseness as defined in Definition 4.1. It is based on the subsumption tree (originally defined only for concept names V_C) which is expanded with negation concepts V_{NC} and specific concepts V_F. The set V_{NC} is formally defined as $V_{NC} = \{ \neg v \mid v \in V_C\}$. Recall that we define $a \not\sqsubseteq b$ for concepts a and b if and only if $\exists y, y \neq \bot$ where y is a concept such that $y \sqsubseteq a \sqcap \neg b$ which involves $\neg b$ and mandates V_{NC}. Even though infinitely many concepts can be constructed from concept names and concept constructors, we only focus on specific concepts V_F which ensure the second requirement in Definition 4.1. A concept $f \in V_F$ corresponds to $f = x \sqcap \neg c$ for $c \in V_C, x \in V_C$ and no proper concept name or concept name negation between $\bot$ and f. Algorithm 4.1 extracts potential evidence by considering nodes along paths in the tree which satisfy both requirements in Definition 4.1. In order for Algorithm 4.1 to check the satisfiability of the requirements, not only V_C but also both V_{NC} and V_F have to be included in the subsumption tree. Hence, the standard tree consisting of concept names V_C only has to be expanded. An algorithm to add V_{NC} and V_F to the existing standard tree is provided in Appendix B.1. The subsumption tree used in Algorithm 4.1 is of the form $\mathcal{G} = (V_C \cup V_{NC} \cup V_F, A)$ where $\mathcal{G}$ is a directed acyclic graph with root $\top$ (top concept). We also define paths and nodes associated with the subsumption tree used in Algorithm 4.1 as follows.

Definition 4.2: $Path_{ij}$ is a set of all possible paths from node j to node i in the subsumption tree. For $path\ P \in Path_{i\top}$ we denote by P_m the m^{th} node in P starting from $\top$. Let $Node_{ij}$ be the set of all nodes along all paths in $Path_{ij}$.

Algorithm 4.1 $(o,\ \bar{O})$ with $\mathcal{G} = (V_C \cup V_{NC} \cup V_F,\ A)$:

---

1 Set $sup_{C(o)} = \emptyset$

2 For each $\bar{o} \in \bar{O}$:

3 Set $sup_{C(\bar{o})} = \emptyset$

4 For each $path\ P \in Path_{C(\bar{o})^\top}$:

5 For $m = length(P)$ to 1:

6 If $P_m \in V_C$:

7 $\Omega_{m,P} = \{y \in V_C \cup V_{NC} | \ y \in Node_{\perp C(o)} \cap Node_{\perp \neg P_m}, y \neq \perp\}$

8 If $\Omega_{m,P} \neq \emptyset$:

9 Add P_m to $sup_{C(\bar{o})}$

10 Else:

11 Break

12 $sup_{C(o)} = sup_{C(o)} \cup sup_{C(\bar{o})}$

13 Remove duplicate nodes in $sup_{C(o)}$

14 Return α = SetCover $(sup_{C(o)})$

In Algorithm 4.1, we assume that $\exists y \in V_C \cup V_{NC}, y \neq \perp$ such that $y \sqsubseteq C(o) \sqcap \neg C(\bar{o})$ for all $\bar{o} \in \bar{O}$ and o in KB. This assumption implies that $C(o)$ cannot be part of $C(\bar{o})$ for an element in $\bar{O}$. If $C(o) \sqsubseteq C(\bar{o})$ for an $\bar{o} \in \bar{O}$, then the statement is true. It assures that $sup_{C(\bar{o})}$ incremented in step 9 for each $\bar{o} \in \bar{O}$ is not empty.

Algorithm 4.1 repeats steps 3-11 to compute potential evidence $\alpha_i's$ for each $\bar{o} \in \bar{O}$ and stores them in $sup_{C(o)}$. All $sup_{C(\bar{o})}'s$ are combined in $sup_{C(o)}$ according to step 12. The set $sup_{C(o)}$ is equivalent to the set of all potential evidences. Note that for every $a \in sup_{C(o)}$ there is $\bar{o}$ such

that $P \in Path_{C(\bar{o})^{\top}}$ contains a due to steps 2-12. Algorithm 4.1 is constructed to ensure that every element in $sup_{C(\mathrm{o})}$ is one of nodes along at least one path from $C(\bar{o})$ to the root. This implies that elements in $sup_{C(\mathrm{o})}$ can be considered as broader concepts of candidate evidence $\bar{o} \in \bar{O}$.

The first requirement in Definition 4.1 requires that any $\bar{o} \in \bar{O}$ has at least one corresponding α_{i} that subsumes $C(\bar{o})$. Hence, the algorithm considers nodes along all possible paths from $C(\bar{o})$ to the root (top concept $\top$) for every $\bar{o} \in \bar{O}$ in order to extract potential evidence α_{i}. The second requirement in Definition 4.1 is directly associated with Ω computed and verified in steps 7 and 8. Both V_{NC} and V_F in the subsumption tree used in Algorithm 4.1 are necessary to compute Ω, i.e., V_{NC} and V_F guarantee that $y \in V_C \cup V_{NC} \cup V_F$. Particularly, $\Omega_{m,P} \neq \emptyset$ implies that $C(o) \not\sqsubseteq P_{\mathrm{m}}$; therefore, P_{m} in this case can be considered as potential evidence α_{i}. Algorithm 4.1 then computes $\Omega_{m,P}$ for each node $P_m \in V_C$ in path $P \in Path_{C(\bar{o})^{\top}}$ corresponding to each $\bar{o} \in \bar{O}$. If $\Omega_{m,P}$ is not empty, P_m (considered as potential evidence α_{i}) is added to $sup_{C(\bar{o})}$ in step 9. Elements in $sup_{C(\bar{o})}$ correspond to nodes in V_C and thus to concept names. They also correspond to potential evidence $\alpha_{\mathrm{i}}'s$. Each $\Omega_{m,P}$ computed in Algorithm 4.1 considers concept names and negation of concept names but Definition 4.1 considers any concept. Algorithm 4.1 consequently provides an approximate evidence set while an exact algorithm is discussed later.

Figure 4.1 illustrates Algorithm 4.1. The path from $C(o)$ to the root $\top$ is highlighted in blue. Nodes along red paths are collected as potential evidence $\alpha_{\mathrm{i}}'s$ as $C(o)$ is not subsumed by these nodes (Ω corresponding to these nodes are not empty).

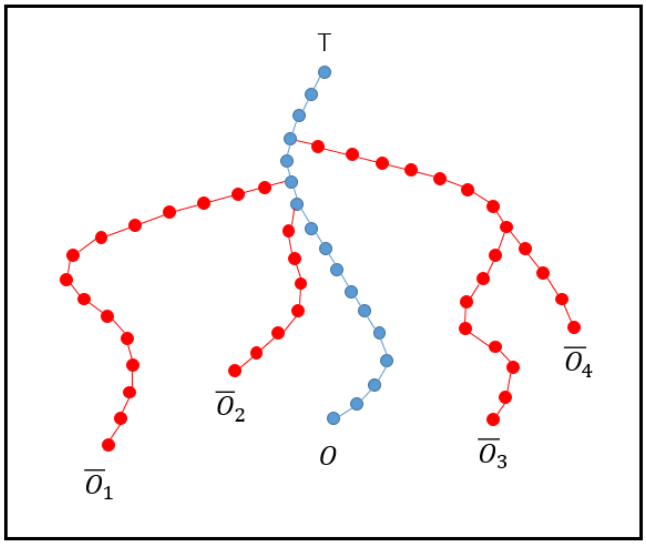

Figure 4.1: Illustration of Algorithm 4.1

Regarding the run time analysis, the proposed algorithm consists of nested loops in steps 2, 4 and 5. Let N be the number of nodes in $\mathcal{G}$ and M the maximum number of paths between any node and the root $\top$. The most outer loop in step 2 considers each element $\bar{o} \in \bar{O}$ which is O(N) while the middle loop in step 4 processes each path $P \in Path_{C(\bar{o})^\top}$ corresponding to $\bar{o}$ in step 2, i.e., O(M). Also, each node along all paths from $C(\bar{o})$ to the root is considered in the most inner loop in step 5, which is O(N). Computing $\Omega_{m,P}$ for each node P_m requires O(N^2). Hence, the computational complexity of Algorithm 4.1 is O($N^4 \cdot M$). Algorithm 4.1 can be sped up by using bisection. The more efficient version is provided in Appendix B.2.

Referring to the running example, we let $o =$ "Minneapolis" and consider $\bar{o} =$ "Mountain View." We consider paths from C("Mountain View") to the root as well as all nodes along these paths. Node $P_m = C$("Mountain View") is added to $sup_{C(\bar{o})}$ as $Node_{\perp C(\text{"Minneapolis"})} \cap Node_{\perp \neg C(\text{"Mountain View"})}$ is not empty. Particularly, there exists a node which belongs to both sets, i.e., $C(\text{"Minneapolis"}) \in Node_{\perp C(\text{"Minneapolis"})}$ and $C(\text{"Minneapolis"}) \in Node_{\perp \neg C(\text{"Mountain View"})}$.

According to a natural geography ontology associated with the example, $sup_{C(\bar{o})} = \{C(\text{“Mountain View”}), C(\text{“Santa Clara”}), C(\text{“California”}), C(\text{“West region”})\}$ is retrieved. Note that $C(\text{“USA”})$ is not in $sup_{C(\bar{o})}$ since $Node_{\perp C(\text{"Minneapolis"})} \cap Node_{\perp \neg C(\text{"USA"})}$ is empty. Intuitively, “Minneapolis” is a location in “USA” (“Minneapolis” is part of “USA”) and therefore, “USA” cannot be counted as an evidence of falseness.

After obtaining the set of all potential α across all possible evidences, we aim to compute an optimal set of evidence with the smallest cardinality. We formally define the object evidence of falseness problem EP as $Z_{EP} = \min_{\alpha \text{ object evidence of falseness}} |\alpha|$. A set covering problem is proposed to find an optimal set of evidence. We later give a condition when it solves it optimally. The set covering problem is formulated as follows.

SetCover($sup_{C(o)}$):

Universe $U = \{\bar{o}_1, \bar{o}_2, \ldots, \bar{o}_{|\bar{O}|}\}$

For any node $a \in sup_{C(o)}$, we define $T_a = \{\bar{o} \in U | a \in P \text{ where } P \in Path_{C(\bar{o})^\top}\}$

The set covering problem SC reads $Z_{SC} = \min |I|$ subject to $\bigcup_{i \in I} T_i = U$. For any node $a \in sup_{C(o)}$, we know that $T_a \subseteq U$ which is necessary for the feasibility of SC. The set covering problem aims to find a minimum number of set $T_a's$ for $a \in sup_{C(o)}$ so that selected sets contain all elements in the universe U, i.e. they cover $\bar{O}$. A feasible solution to the set covering problem satisfies the first requirement of Definition 4.1. The set T_a for each $a \in sup_{C(o)}$ is specifically constructed based on $sup_{C(o)}$. Note that the set $T_a \neq \emptyset$ because of the fact that for every $a \in$

$sup_{C(o)}$ there is $\bar{o}$ such that $P \in Path_{C(\bar{o})^{\top}}$ contains a and the construction of T_a. All elements in $sup_{C(o)}$ added in Algorithm 4.1 are guaranteed to satisfy the second requirement of Definition 4.1.

According to the false triplet ("Google", "OfficeLocationInUS", "Minneapolis") example, we consider $o =$ "Minneapolis" and the set $\bar{O}$ given previously. A set of generated $T_a's$ which yields a feasible solution to the set covering problem is given in Table 4.1. The left column lists 5 elements from $sup_{C(o)}$.

Table 4.1. An example of feasible $T_a's$ sets for the set covering problem

$T_{C(\text{"West region"})}$	"Boulder," "Irvine," "Kirkland," "Los Angeles," "Mountain View," "Playa Vista," "San Bruno," "San Francisco," "Seattle," "Sunnyvale"
$T_{C(\text{"Northeast region"})}$	"Cambridge," "New York," "Pittsburgh"
$T_{C(\text{"South region"})}$	"Atlanta," "Austin," "Chapel Hill," "Miami," "Reston," "Washington DC"
$T_{C(\text{"Michigan"})}$	"Ann Arbor," "Birmingham"
$T_{C(\text{"Illinois"})}$	"Chicago"

Propositions 4.1 and 4.2 stated next establish the relationship between EP and SC. Proofs of Propositions 4.1 and 4.2 are provided in Appendix B.3.

Proposition 4.1: SC is feasible and a feasible solution to SC yields a feasible solution to EP of same or smaller cardinality.

This implies that $Z_{SC} \geq Z_{EP}$. Due to Proposition 4.1, a solution to SC is always a feasible solution to EP and thus an object evidence of falseness obtained from SC can be used as a representative of the evidence set from EP.

We consider either concept names or negation of concept names when $\Omega_{m,P}$ is computed in Algorithm 4.1. An exact algorithm replaces $\Omega_{m,P}$ defined in step 7 of Algorithm 4.1 with $\Omega'_{m,P} = \{\text{concept } y \,|y \sqsubseteq C(o) \sqcap \neg P_i \,, y \neq \bot\}$. All concepts constructed from concept names and concept constructors are considered in $\Omega'_{m,P}$. We also observe that checking $\Omega'_{m,P} \neq \emptyset$ is equivalent to checking satisfiability of the concept $C(o) \sqcap \neg P_m$, i.e. if $C(o) \sqcap \neg P_m$ is satisfiable, then $\Omega'_{m,P} \neq \emptyset$ as stated in [91].

Proposition 4.2: If $\Omega_{m,P}$ in step 7 of Algorithm 4.1 is substituted with $\Omega'_{m,P}$, then $Z_{SC} = Z_{EP}$.

In the proof for Proposition 4.2, we show that a feasible solution to EP is also a feasible solution to SC when replacing $\Omega_{m,P}$ with $\Omega'_{m,P}$. This implies that $Z_{EP} \geq Z_{SC}$ and combined with Proposition 4.1 it yields $Z_{SC} = Z_{EP}$.

We define a subject evidence $\beta = \{\beta_1 \,, \beta_2, \dots, \beta_k\}$ in the same way. In order to identify the subject evidence of falseness β, Algorithm 4.1 is applied where all definitions and propositions are defined similarly with respect to $(\bar{s}, r, \mathrm{o})$. A domain under consideration can have multiple ontologies. In such a case, we implement the proposed algorithms to identify the evidence of falseness for each ontology. The minimum cardinality of subject/object evidence is selected across all ontologies. Finally, the problem to identify the evidence of falseness for each triplet (s, r, o) is

formally formulated by considering both subject and object evidence sets as $\min\left\{\min_{ont \in \text{ontologies}}\left(\min_{\alpha \text{ object evidence in } ont}|\alpha|\right)_{ont}, \min_{ont \in \text{ontologies}}\left(\min_{\beta \text{ subject evidence in } ont}|\beta|\right)_{ont}\right\}$.

4.4 Case Study

We apply the proposed algorithm to the health care domain as a case study. A reliable source in our case is obtained from biomedical publications stored in the MEDLINE database. In order to construct KG, SemRep [90] is used to extract semantic predicate triplets from biomedical texts. SemRep matches subject and object entities in triplets with concepts from the UMLS Metathesaurus and matches relationship with respect to the UMLS Semantic Network. It also takes into account a syntactic analysis, a structured domain knowledge, and hypernymic propositions extensively. The data contains both the extracted triplet and the corresponding sentence from MEDLINE.

We first train the PRA model based on KG constructed from SemRep. We further compare its performance with the evaluation metrics reported in [93] where PRA has been trained on the NELL data set. The average mean reciprocal rank (MRR) across different relation types reported in [93] is 0.516 while the average MRR of PRA on SemRep is 0.25. The MRR is computed based on the rank of the first correctly retrieved triplet; however we aim to correctly retrieve all triplets that are in KG. As a result, we additionally compute the mean average precision (MAP) which considers the rank position of each triplet in KG. The MAP based on our trained PRA model is 0.1. This implies that on average every 10th retrieved result is correct. We then manually inspect the original statements and their corresponding predicate triplets extracted from SemRep. Even

though a preliminary evaluation of SemRep reported in [90] states 83% precision, extracted predicate triplets in KG contain many errors based on our manual observation. Examples of predicate triplets incorrectly extracted from original statements are provided in Appendix B.4. The issue is that the sentences are clearly correct but the extracted triplets are often wrong.

Hence, we pre-process KG by verifying each extracted predicate triplet with the PRA model and additional relation extraction systems. Detailed explanations are provided in the following data preparation section.

4.4.1 Data Preparation

We aim to re-construct KG containing only triplets with high precision. After manually observing results from the trained PRA model, triplets with high PRA scores tend to be more accurate than those with low PRA scores. Hence, PRA is one of models used to verify triplets in KG. We further employ other relation extraction systems to filter out incorrect triplets from the original KG. Ollie [94] is an open information extraction software which aims to extract binary relationships from sentences. According to open information extraction, a schema of relations does not need to be pre-specified. In addition, we train a recurrent neural network model called LSTM-ER proposed by Miwa and Bansal [95] on a publicly available training data set having gold standard labels. Each instance in the training data consists of a statement and its predicate triplet. The training data set used to train the LSTM-ER model includes the ADE corpus [96], SemEval-2010 [97], BioNLP [98], and the SemRep Gold standard annotation [99].

In order to pre-process the original KG, we propose a strategy to combine triplets with high PRA scores and triplets matching with the Ollie or LSTM-ER models. A flow diagram of the proposed strategy in order to construct an adjusted KG is depicted in Figure 4.2.

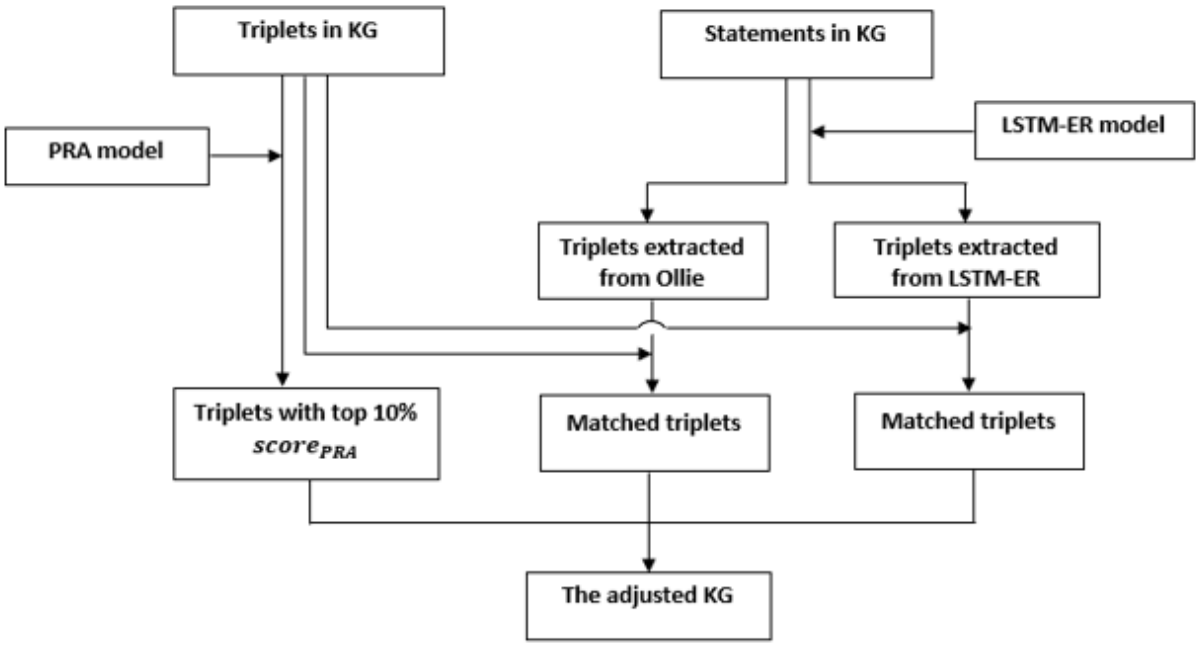

Figure 4.2: The flow diagram of the strategy to pre-process KG

According to the proposed strategy, we infer the trained PRA model on KG and rank the results from high to low PRA scores. The triplets positioned in the top 10 percent of the ranked PRA scores are retrieved. Additionally, we collect all possible matches between triplets in KG and results from Ollie. We also use the trained LSTM-ER model to infer possible relations from statements associated with triplets in the original KG. Each triplet from KG is collected if its relation matches with the relation inferred from the LSTM-ER model. We conduct a preliminary experiment by extracting matched triplets using Ollie, and the LSTM-ER model based on 10,000

randomly selected triplets. Based on the experiment, we observe a small proportion of matching triplets among different relation extraction models. A detailed discussion of the experiment is provided in Appendix B.5.

Additionally, we observe that many statements in the original KG involve studies with non-human subjects such as “Effects of acetylcholine, histamine, and serotonin infusion on venous return in dogs.” In order to filter out these statements, we consider the UMLS semantic type, a categorization of concepts represented in the UMLS Metathesaurus, tagged in the statements. In particular, we eliminate statements which contain “Amphibian,” “Animal,” “Bird,” “Fish,” “Mammal,” “Reptile,” and “Vertebrate” semantic types. We provide the number of nodes and edges in the original KG and in the adjusted KG in Table 4.2. The average MRR and the average MAP based on the PRA model on the adjusted KG are 0.44 and 0.29, respectively

Table 4.2: Number of nodes and edges in the original KG and the adjusted KG

	The original KG	**The adjusted KG**
Number of nodes	229,063	161,930
Number of edges	15,700,435	4,107,296

4.4.2 Results

We run the whole pipeline of Algorithm 4.1 to validate the truthfulness and provide supporting evidence of lay triplets with the adjusted KG. Based on 2,084 lay triplets consisting of 20 relation types collected from health-related web pages, we identify the truthfulness of each

triplet and extract evidence candidates as summarized in Appendix B.6. Across all relation types, there are 501 false triplets which account for 24 percent of all lay triplets.

Instead of directly specifying thresholds ε_1 and ε_2 in step 1 of the process, we identify the rank threshold $rank_{\varepsilon_1}$ and $rank_{\varepsilon_2}$ based on the ordered PRA scores. To identify $rank_{\varepsilon_1}$ corresponding to subject entity s and relation type r, we retrieve a set of all object entities which have relation $r(\mathrm{s},\cdot)$ identified by the PRA model. This set is denoted as O_{all}. The set $O_{KG} = \{\tilde{o} \,|(s,r,\tilde{o}) \in KG\}$ is also retrieved. A parameter x which is defined as $x = \mathbf{1}_{|O_{KG}|>0} \frac{|O_{KG}|}{median(PRA\ scores\ of\ O_{KG})}$ is used to specify the rank threshold $rank_{\varepsilon_1}$ as follows:

$$rank_{\varepsilon_1} = \begin{Bmatrix} 5 & \text{if} & x \leq 0.25 \\ 10 & \text{if} & 0.25 < x \leq 0.5 \\ 15 & \text{if} & 0.5 < x \leq 0.75 \\ 20 & \text{if} & x > 0.75 \end{Bmatrix} + 5 - \frac{|O_{all}|}{10000}.$$

The parameter x captures how well the PRA model gives high ranks to triplets in KG. The higher the value of x is, the better the PRA model performs. This implies that $rank_{\varepsilon_1}$ should vary proportionally to x. The middle term is a hyper parameter calibrated in the experiment in order to obtain the best performance. The last term in the $rank_{\varepsilon_1}$ formula takes into account how $|O_{all}|$ affects $rank_{\varepsilon_1}$. Having high $|O_{all}|$ indicates that many object entities are predicted to have $r(\mathrm{s},\cdot)$ with respect to subject s. Therefore, it is more challenging for the PRA model to correctly rank retrieved object entities. This implies that high $|O_{all}|$ leads to low value of the threshold $rank_{\varepsilon_1}$ as expressed in the formula. We extract entity $\bar{o}$ whose rank based on PRA score is higher than $rank_{\varepsilon_1}$ as candidates in $\bar{O}$. Moreover, we specify $rank_{\varepsilon_2}$ as $\max(rank_{\varepsilon_1},\ 0.005 * O_{all})$ to identify the truthfulness of (s,r,o). If the rank of $score_{PRA}(\mathrm{o};\mathrm{s})$ is higher than $rank_{\varepsilon_2}$, we specify (s,r,o) as true.

Among 501 false lay triplets, we first eliminate triplets whose object o does not match with concepts in ontologies. Candidates are then used to compute the evidence set based on the remaining 395 triplets by using Algorithm 4.1. We only perform evaluations on object candidates while subject candidates can be done similarly. The average cardinality of object candidates $|\overline{O}|$ and the average cardinality of their corresponding evidence sets $|\alpha|$ across all relation types are 11.65 and 2.24, respectively. A histogram of the produced object evidence $|\alpha|$ of all relation types based on candidates $\overline{O}$ is provided in Figure 4.3.

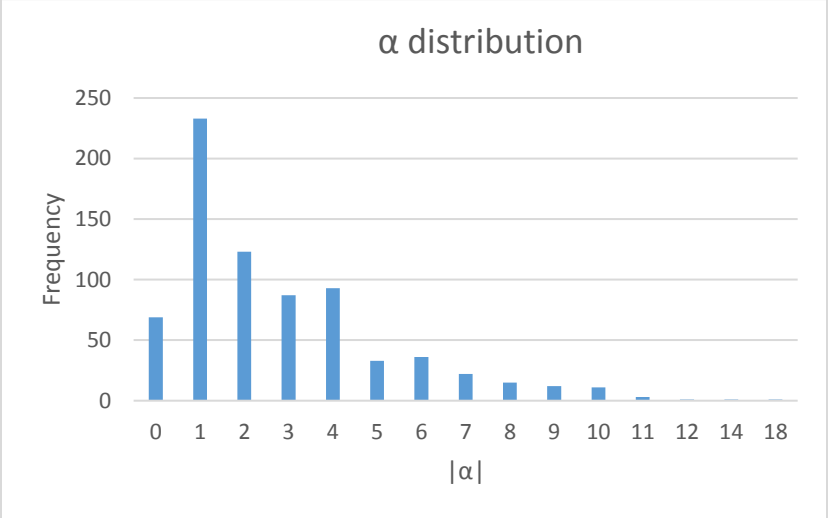

Figure 4.3: A histogram of object evidence $|\alpha|$

In order to evaluate the performance of the proposed algorithm, we choose 3 relation types representing high, medium and low MAP computed from the PRA model which are "TREATS," "DIAGNOSES," and "CAUSES," respectively. For each relation type, we select 5 cases to compare the evidence sets resulting from the algorithm (denoted as "Al") against the evidence sets

constructed manually (denoted as "Ma") as illustrated in Table 4.3. A complete comparison of elements in evidence sets "Al" and "Ma" is provided in Appendix B.7.

Table 4.3: Evidence sets obtained from the algorithm and the evidence sets constructed manually

False triplets	**|Al|**	**|Ma|**	**|Al ∩ Ma|**	**|Al\Ma|**	**|Ma\Al|**
Relation type: TREATS					
Heparin TREATS Fever	1	1	1	0	0
Amiodarone TREATS Hepatitis C	1	1	1	0	0
Stress management TREATS Mitral Valve Prolapse	1	1	1	0	0
Capoten TREATS Coughing	1	1	1	0	0
Losartan TREATS Varicose Ulcer	3	2	1	2	1
Average of TREATS	1.4	1.2	1.0	0.4	0.2
Relation type: DIAGNOSES					
Echocardiography DIAGNOSES Hyperlipidemia	1	1	1	0	0
Platelet Size DIAGNOSES Anemia	1	2	1	0	1
Esophageal pH Monitoring DIAGNOSES Malignant breast neoplasm	5	3	3	2	0
Cholesterol measurement test DIANOSES Malignant breast neoplasm	5	2	2	3	0
Electrocardiogram DIAGNOSES Muscle strain	1	1	1	0	0
Average of DIAGNOSES	2.6	1.8	1.6	1.0	0.2
Relation type: CAUSES					

Caffeine CAUSES Gout	1	3	1	0	2
hypercholesterolemia CAUSES Neuropathy	2	2	2	0	0
Leukemia CAUSES Gout	2	2	2	0	0
Harpin CAUSES Cardiomegaly	2	1	1	1	0
Ascorbic Acid CAUSES Senile Plaques	2	2	1	1	1
Average of CAUSES	1.8	2.0	1.4	0.4	0.6
Average across 3 relation types	1.9	1.7	1.3	0.6	0.3

4.5 CONCLUSIONS AND FUTURE WORK

In this work, we develop a system to validate the truthfulness of lay triplets and provide supporting evidence. Our system employs the PRA algorithm inferred on KG re-constructed from reliable sources to identify whether a lay triplet is true or false. In our experiment, we train the PRA model based on KG constructed from biomedical literature. The original KG contains incorrect triplets due to the relation extraction process. We attempt to re-construct KG consisting of more accurate triplets by verifying each triplet in the original KG with additional relation extraction algorithms. The trained PRA model on the adjusted KG yields 0.44 MRR and 0.29 MAP averaged across all relation types. The performance of the PRA model based on the adjusted KG is improved. However, the adjusted KG still contains errors due to the challenge of complicated biomedical text and limited resources in training additional relation extraction algorithms.

We use a combination of knowledge from ontologies and triplets in the adjusted KG to extract a concise supporting evidence set. Specifically, Algorithm 4.1 aims to find the supporting

evidence set which does not overlap with an entity in a lay triplet. The evidence set is aggregated from candidates obtained from triplets in the adjust KG by using knowledge of ontologies. We apply Algorithm 4.1 to extract evidence sets based on each ontology and repeatedly consider all possible ontologies. It is reasonable to select the ontology which yields the minimum cardinality of evidence sets. According to our algorithm, we first match object (subject) entity in a lay triplet with concepts within all ontologies. Non-matching ontologies are not taken into a consideration when object (subject) candidates are paired with concepts in ontologies. We assume that candidates that cannot be matched with concepts in the same ontology as the lay triplet's are disregarded.

We consider the number of candidates $|\overline{O}|$ extracted from the PRA model and compare it against the cardinality of the evidence set $|\alpha|$ resulted from the algorithm. The average of $|\overline{O}|$ is larger than the average of $|\alpha|$ by a factor of 5 across all relation types. This implies that our proposed algorithm provides valid and concise evidence sets. To evaluate the performance of our algorithm, we compare the evidence set extracted from our proposed algorithm with a manually-constructed evidence set. The average number of overlap between the evidence set from the algorithm and the manually constructed set is 74% across the 3 relation types. Our proposed algorithm performs very well especially with some specific relation types such as "TREATS" with the overlap of 87%.

The problem is challenging due to limited resources to construct a complete and accurate KG. An imperfect KG plays a significant role in the inferior performance of the PRA model which directly impacts the performance of Algorithm 4.1 to extract evidence sets. A better quality of KG would lead to a higher performance of the proposed system. Hence, future work should focus on

improving relation extraction algorithms to construct KG. We believe that this is of utmost importance, not just for our work, but all systems that rely on SemRep.

References

1. Powers, D.M.W., *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation.* Journal of machine learning technologies, 2011. **2**(1): p. 37-63.
2. Chien, C.V., *Predicting patent litigation.* Texas law review, 2011. **90**(2): p. 283-329.
3. Petherbridge, L., *On predicting patent litigation.* Texas law review, 2012. **90**(75): p. 75-86.
4. Kesan, J.P., D.L. Schwartz, and T.M. Sichelman, *Paving the path to accurately predicting legal outcomes: A comment on professor Chien's predicting patent litigation.* Texas law review, 2012. **90**: p. 97-109.
5. Kacsuk, Z., *The mathematics of patent claim analysis.* Artificial intelligence and law, 2011. **19**(4): p. 263-289.
6. Su, H.N., C.M.L. Chen, and P.N. Lee, *Patent litigation precaution method: analyzing characteristics of US litigated and non-litigated patents from 1976 to 2010.* Scientometrics, 2012. **92**(1): p. 181-195.
7. Lanjouw, J.O. and M. Schankerman, *Characteristics of patent litigation: A window on competition.* Rand journal of economics, 2001. **32**(1): p. 129-151.
8. Lanjouw, J.O. and M. Schankerman, *Protecting intellectual property: Are small firms handicapped?* Journal of law and economics, 2004. **47**(1): p. 45-74.
9. Cremers, K., *Determinants of patent litigation in Germany.* ZEW - Center for European economic research discussion paper, 2004(04-072).

10. Ashley, K.D. and S. Bruninghaus, *Automatically classifying case texts and predicting outcomes.* Artificial intelligence and law, 2009. **17**(2): p. 125-165.

11. Cowart, T.W., R. Lirely, and S. Avery, *Two methodologies for predicting patent litigation outcomes: Logistic regression versus classification trees.* American business law journal, 2014. **51**(4): p. 843-877.

12. Kashima, H., et al. *Predicting modeling of patent quality by using text mining*. in *International association for management of technology*. 2010. Cairo, Egypt.

13. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey.* ACM Computing surveys 2009. **41**(3): p. 1-58.

14. Nalbantov, G.I., P.J.F. Groenen, and J.C. Bioch, *Nearest convex hull classification*. 2006.

15. Chawla, N.V., et al., *SMOTE: Synthetic minority over-sampling technique.* Journal of artificial intelligence research, 2002. **16**(1): p. 321-357.

16. Vens, C., et al., *Decision trees for hierarchical multi-label classification.* Machine learning 2008. **73**(2): p. 185-214.

17. Fox, S. and M. Duggan, *Health online 2013*. 2013, PewResearch center: http://www.webcitation.org/6eAaGTAoU

18. Lau, A. and E. Coiera, *Impact of web searching and social feedback on consumer decision making: A prospective online experiment.* Journal of medicine internet research, 2008. **10**(1): p. e2.

19. Nath, C., et al., *Analysis of website sharing in online health communities.* Journal of medical internet research, 2016. **18**(1): p. e11.

20. Del Fiol, G., et al. *"On-demand" access to a multi-purpose collection of best practice standards*. in *Conference proceedings IEEE Engineering in medicinea nd biology society*. 2004.

21. Del Fiol, G., et al., *An XML model that enables the development of complex order sets by clinical experts.* IEEE Transactions on information technology in biomedicine, 2005. **9**(2): p. 216-228.

22. Del Fiol, G., R.A. Rocha, and P.D. Clayton. *Infobuttons at intermountain healthcare: Utilization and infrastructure*. in *AMIA Annual symposium proceedings*. 2006.

23. Del Fiol, G., et al. *Integrating genetic information resources with an EHR*. in *AMIA Annual symposium proceedings*. 2006.

24. Cimino, J.J. and G. Del Fiol, *Infobuttons and point of care access to knowledge*, in *Clinical decision support: the road ahead*, R.A. Greenes, Editor. 2007, Academic Press: Boston, MA.

25. Del Fiol, G. and P.J. Haug. *Use of classification models based on usage data for the selection of infobutton resources*. in *AMIA Annual symposium proceedings*. 2007.

26. Del Fiol, G. and P.J. Haug, *Infobuttons and classification models: A method for the automatic selection of on-line information resources to fulfill clinicians' information needs.* Journal of biomedical informatics, 2008. **41**(4): p. 655-666.

27. Del Fiol, G., et al., *Effectiveness of topic-specific infobuttons: A randomized controlled trial.* Journal of the American medical informatics association, 2008. **15**(6): p. 752-9.

28. Del Fiol, G. and P.J. Haug, *Classification models for the prediction of clinicians' information needs.* Journal of biomedical informatics, 2009. **42**(1): p. 82-89.

29. Del Fiol, G., et al. *A large-scale knowledge management method based on the analysis of the use of online knowledge resources*. in *AMIA Annual symposium proceedings*. 2010.

30. Del Fiol, G., et al., *Implementations of the HL7 context-aware knowledge retrieval ("Infobutton") standard: Challenges, strengths, limitations, and uptake.* Journal of biomedical informatics, 2012.

31. Jonnalagadda, S.R., et al., *Automatically extracting sentences from Medline citations to support clinicians' information needs.* Journal of the American medical informatics association 2012.

32. Mishra, R., et al. *Automatically extracting clinically useful sentences from UpToDate to support clinicians' information needs*. in *AMIA Annual symposium proceedings*. 2013.

33. Zhang, M., et al., *Automatic identification of comparative effectiveness research from medline citations to support clinicians' treatment information needs.* Studies in health technology and informatics, 2013. **192**: p. 846-50.

34. Mishra, R., et al., *Text summarization in the biomedical domain: A systematic review of recent research.* Journal of biomedical informatics, 2014. **52**: p. 457-467.

35. Bui, D.D., S. Jonnalagadda, and G. Del Fiol, *Automatically finding relevant citations for clinical guideline development.* Journal of biomedical informatics, 2015.

36. Morid, M., et al. *Classification of clinically useful sentences in MEDLINE*. in *AMIA Annual symposium proceedings* 2015.

37. Barnett, G.O., et al., *An evolving diagnostic decision-support system.* Journal of American medical association, 1987. **258**: p. 67-74.

38. Cimino, J.J., G. Elhanan, and Q. Zeng. *Supporting infobuttons with terminological knowledge*. in *AMIA Annual fall symposium*. 1997.

39. Currie, L.M., et al. *Clinical information needs in context: An observational study of clinicians while using a clinical information system*. in *AMIA Annual symposium proceedings*. 2003.

40. Janetzki, V., M. Allen, and J.J. Cimino, *Using natural language processing to link from medical text to on-line information resources*, in *MEDINFO*. 2004.

41. Cimino, J.J. *Use, usability, usefulness, and impact of an infobutton manager*. in *AMIA Annual symposium proceedings*. 2006.

42. Cimino, J., *Infobuttons: Anticipatory passive decision support.*, in *AMIA Annual symposium proceedings*. 2008. p. 1203.

43. Cimino, J.J., *The contribution of observational studies and clinical context information for guiding the integration of Infobuttons into clinical information systems*, in *AMIA Annual symposium proceedings*. 2009, American medical informatics association. p. 109.

44. Collins, S.A., et al., *Information needs, Infobutton manager use, and satisfaction by clinician type: a case study.* Journal of the American medical informatics association, 2009. **16**: p. 140.

45. Cao, Y., et al., *AskHERMES: An online question answering system for complex clinical questions.* Journal of biomedical informatics, 2011. **44**: p. 277-88.

46. Huser, V. and J. Cimino, *Evaluating adherence to the international committee of medical journal editors' policy of mandatory, timely clinical trial registration.* Journal of the American medical informatics assocociation, 2013. **20**: p. e169 - e174.

47. Athenikos, S.J. and H. Han, *Biomedical question answering: A survey.* Computer methods and programs in biomedicine, 2010. **99**(1): p. 1-24.

48. Shtok, A., et al., *Learning from the past: Answering new questions with past answers*, in *Proceedings of the 21st international conference on world wide web*. 2012: Lyon, France p. 759-768.

49. Marom, Y. and I. Zukerman, *A predictive approach to help-desk response generation*, in *20th international joint conference on artificial intelligence*. 2007: Hyderabad, India p. 1665-1670.

50. Feng, D., et al., *An intelligent discussion-bot for answering student queries in threaded discussions*, in *Proceedings of the 11th international conference on intelligent user interfaces*. 2006: Sydney, Australia p. 171-177.

51. Wang, K., Z.Y. Ming, and T.S. Chua, *A Syntactic tree matching approach to finding similar questions in community-based QA services*, in *Proceedings 32nd annual international ACM SIGIR conference on research and development in information retrieval*. 2009: Boston, Massachusetts, USA. . p. 187-194.

52. Jeon, J., W.B. Croft, and J.H. Lee, *Finding similar questions in large question and answer archives*, in *Proceedings of the 14th ACM international conference on information and knowledge management*. 2005: Bremen, Germany p. 84-90.

53. Bernhard, D. and I. Gurevych, *Answering learners' questions by retrieving question paraphrases from social Q&A sites*, in *Proceedings of the third workshop on innovative use of NLP for building educational applications*. 2008: Columbus, Ohio, USA. . p. 44-52.

54. Zhang, W.N., et al., *A topic clustering approach to finding similar questions from large question and answer archives.* Plos one, 2014. **9**(3): p. e71511.

55. Ko, J., L. Si, and E. Nyberg, *A probabilistic framework for answer selection in question answering*, in *Proceedings of human language technology conference of the North*

American chapter of the association of computational linguistics. 2007: Rochester, New York, USA. . p. 524-531.

56. Moschitti, A. and S. Quarteroni, *Linguistic kernels for answer re-ranking in question answering systems.* Information processing & management, 2011. **47**(6): p. 825-842.

57. Suzuki, J., Y. Sasaki, and E. Maeda, *SVM answer selection for open-domain question answering*, in *Proceedings of the 19th international conference on computational linguistics*. 2002: Taipei, Taiwan p. 1-7.

58. Blooma, M.J., A.Y.K. Chua, and D.H.L. Goh, *Selection of the best answer in CQA services*, in *Proceedings of the 7th international conference on information technology*. 2010: Las Vegas, NV, USA. . p. 534-539.

59. Wu, Y., et al., *Learning unsupervised SVM classifier for answer selection in web question answering*, in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*. 2007: Prague, Czech Republic. p. 33-41.

60. Toba, H., et al., *Discovering high quality answers in community question answering archives using a hierarchy of classifiers.* Information sciences, 2014. **261**: p. 101-115.

61. Shah, C. and J. Pomerantz, *Evaluating and predicting answer quality in community QA*, in *SIGIR 2010: Proceedings of the 33rd annual international ACM SIGIR conference on research development in information retrieval*. 2010: Geneva, Switzerland p. 411-418.

62. Arai, K. and A.N. Handayani, *Predicting quality of answer in collaborative Q/A community.* International journal of advanced research in artificial intelligence, 2013. **2**(3): p. 21-25.

63. Shah, C., V. Kitzie, and E. Choi, *Questioning the question - addressing the answerability of questions in community question-answering*, in *2014 47th Hawaii international conference on system sciences* 2014: Waikoloa, HI, USA. . p. 1386-1395.

64. Tian, Q., P. Zhang, and B. Li, *Towards predicting the best answers in community-based question-answering services*, in *Proceedings of the 7th international conference on weblogs and social media*. 2013: Cambridge, MA, USA. . p. 725-728.

65. Luo, J., et al., *SimQ: Real-time retrieval of similar consumer health questions.* Journal of medicine internet research, 2015. **17**(2): p. e43.

66. *NetWellness*. http://www.webcitation.org/6eJJdNX5W.

67. Wong, W., J. Thangarajah, and L. Padgham, *Contextual question answering for the health domain.* Journal of the american society for information science and technology, 2012. **63**(11): p. 2313-2327.

68. Nigam, K., et al., *Text classification from labeled and unlabeled documents using EM.* Machine learning, 2000. **39**(2-3): p. 103-134.

69. Liu, X.Y., Y.M. Zhou, and R.S. Zheng, *Sentence similarity based on dynamic time warping*, in *International conference on semantic computing, proceedings*. 2007: Irvine, CA, USA. . p. 250-256.

70. Zobel, J. and A. Moffat, *Exploring the similarity space.* ACM SIGIR Forum 1998. **32**(1): p. 18-34.

71. Levenshtein, V., *Binary codes capable of correcting deletions, insertions, and reversals.* Soviet physics-Doklady, 1966. **10**(8): p. 707-710.

72. Wagner, R.A. and M.J. Fischer, *The string-to-string correction problem.* Journal of ACM, 1974. **21**(1): p. 168-173.

73. Kecman, V., *Learning and soft computing : support vector machines, neural networks, and fuzzy logic models*. Complex adaptive systems. 2001, Cambridge, Mass.: MIT Press. xxxii, 541 p.

74. Wongchaisuwat, P., D. Klabjan, and S. Jonnalagadda, *A Semi-Supervised Learning Approach to Enhance Community-based Question Answering: Code and Dataset*. 2015: https://github.com/papisw/Health-QA.

75. Anderson, M. and A. Perrin *13% of Americans don't use the internet. Who are they?* 2016.

76. Gottleb, S., *Health information on internet is often unreliable.* British medical journal, 2000. **321**(7254): p. 136.

77. Waguih, D.A. and L. Berti-Equille, *Truth discovery algorithms: An experimental evaluation.* arXiv preprint arXiv:1409.6428, 2014.

78. Li, Q., et al., *A confidence-aware approach for truth discovery on long-tail data.* Proceedings very large data base endowment, 2014. **8**(4): p. 425-436.

79. Ma, F., et al., *FaitCrowd: Fine grained truth discovery for crowdsourced data aggregation*, in *Proceedings of the 21th ACM SIGKDD International conference on knowledge discovery and data mining*. 2015, ACM: Sydney, NSW, Australia. p. 745-754.

80. Meng, C., et al., *Truth discovery on crowd sensing of correlated entities*, in *Proceedings of the 13th ACM Conference on embedded networked sensor systems*. 2015, ACM: Seoul, South Korea. p. 169-182.

81. Mukherjee, S., G. Weikum, and C. Danescu-Niculescu-Mizil, *People on drugs: Credibility of user statements in health communities*, in *Proceedings of the 20th ACM SIGKDD International conference on knowledge discovery and data mining*. 2014, ACM: New York, USA. p. 65-74.

82. Xiao, H., et al., *Towards confidence in the truth: A bootstrapping based truth discovery approach*, in *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining*. 2016, ACM: San Francisco, California, USA. p. 1935-1944.

83. Zhao, Z., J. Cheng, and W. Ng, *Truth discovery in data streams: A single-pass probabilistic approach*, in *Proceedings of the 23rd ACM International conference on conference on information and knowledge management*. 2014, ACM: Shanghai, China. p. 1589-1598.

84. Zhi, S., et al., *Modeling truth existence in truth discovery*, in *Proceedings of the 21th ACM SIGKDD International conference on knowledge discovery and data mining*. 2015, ACM: Sydney, NSW, Australia. p. 1543-1552.

85. Liu, G., et al., *TruthOrRumor: Truth judgment from web*, in *web technologies and applications: 16th Asia-Pacific web conference, APWeb 2014, Changsha, China, September 5-7, 2014. Proceedings*, L. Chen, et al., Editors. 2014, Springer International Publishing: Cham. p. 674-678.

86. Wang, T., Q. Zhu, and S. Wang, *Multi-verifier: A novel method for fact statement verification*, in *web technologies and applications: 15th Asia-Pacific web conference, APWeb 2013, Sydney, Australia, April 4-6, 2013. Proceedings*, Y. Ishikawa, et al., Editors. 2013, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 526-537.

87. Li, X., W. Meng, and C. Yu. *T-verifier: Verifying truthfulness of fact statements*. in *2011 IEEE 27th International conference on data engineering*. 2011.

88. Yin, X. and W. Tan, *Semi-supervised truth discovery*, in *Proceedings of the 20th International conference on world wide web*. 2011, ACM: Hyderabad, India. p. 217-226.

89. Yamamoto, Y. and K. Tanaka, *Finding comparative facts and aspects for judging the credibility of uncertain facts*, in *web information systems engineering - WISE 2009: 10th International conference, Poznań, Poland, October 5-7, 2009. Proceedings*, G. Vossen, D.D.E. Long, and J.X. Yu, Editors. 2009, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 291-305.

90. Rindflesch, T.C. and M. Fiszman, *The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text.* Journal of biomedical informatics, 2003. **36**(6): p. 462-477.

91. Franz, B., et al., eds. *The description logic handbook: theory, implementation, and applications*. 2003, Cambridge University Press. 545.

92. Baader, F., et al., *An empirical analysis of optimization techniques for terminological representation systems.* Applied intelligence, 1994. **4**(2): p. 109-132.

93. Lao, N., T. Mitchell, and W.W. Cohen, *Random walk inference and learning in a large scale knowledge base*, in *Proceedings of the conference on empirical methods in natural language processing*. 2011, Association for computational linguistics: Edinburgh, United Kingdom. p. 529-539.

94. Mausam, et al., *Open language learning for information extraction*, in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, Association for computational linguistics: Jeju Island, Korea. p. 523-534.

95. Miwa, M. and M. Bansal, *End-to-end relation extraction using lstms on sequences and tree structures.* arXiv preprint arXiv:1601.00770, 2016.

96. Gurulingappa, H., et al., *Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports*. Journal of biomedical informatics, 2012. **45**(5): p. 885-892.

97. Hendrickx, I., et al., *SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals*, in *Proceedings of the 5th International workshop on semantic evaluation*. 2010, Association for computational linguistics: Los Angeles, California. p. 33-38.

98. Kim, J.-D., et al., *Overview of BioNLP shared task 2011,* in *Proceedings of the BioNLP shared task 2011 workshop*. 2011, Association for computational linguistics: Portland, Oregon. p. 1-6.

99. Kilicoglu, H., et al., *Constructing a semantic predication gold standard from the biomedical literature*. BMC Bioinformatics, 2011. **12**(1): p. 486.

Appendix A

Table A.1 Selected lists of patent classes[5] corresponding to the 3 different keywords

Keyword	Class number
Wireless Network	235, 340, 342, 343, 370, 375, 379, 380, 455, 463, 700, 701, 702, 704, 705, 706, 707, 709, 713, 714, 715, 716, 717, 725, 726
Advertising	2, 40, 52, 53, 156, 198, 206, 211, 220, 229, 235, 248, 257, 273, 280, 313, 340, 345, 347, 348, 358, 359, 362, 370, 379, 382, 386, 424, 428, 430, 435, 446, 455, 463, 473, 600, 700, 701, 702, 704, 705, 706, 707, 709, 713, 714, 715, 717, 725, 726, D20
Telecommunication	174, 235, 327, 333, 340, 341, 342, 343, 348, 356, 358, 359, 361, 370, 372, 375, 379, 382, 385, 398, 439, 455, 600, 701, 702, 704, 705, 707, 709, 710, 713, 714, 715, 725, 726

[5] Corresponding class title to class number can be viewed at https://www.uspto.gov/web/patents/classification/selectnumwithtitle.htm

Appendix B

B.1 Algorithm to add negation nodes to the subsumption tree

Algorithm B1 presented next adds negation nodes V_{NC} to the existing subsumption tree by implementing the incremental methods defined in [92]. An augmented tree is of the form $\mathcal{G} = (V_C \cup V_{NC} \cup V_F\,,\ A)$ where nodes $v \in V_C$ are concept names, nodes $v \in V_{NC}$ are negation concepts of concept names and nodes $v \in V_F$ are specific concepts which ensure the second requirement in Definition 4.1. Arc $(b, a) \in A$ between node a and b has the following properties.

1. $a \in V_C \cup V_{NC}$ and $b \in V_C \cup V_{NC} => \ a \prec b$, i.e. $a \leq b$ and if there exists $z \in V_C \cup V_{NC}$ such that $a \leq z \ \leq b$, then $z = a$ or $z = b$
2. $a \in V_F$ and $b \in V_C \cup V_{NC} => a \leq b$

Concept names and concept constructors are combined in order to construct concepts. This consequently yields a significant number of concepts. Nodes in V_F may have $\leq$ relation with many concepts which are not added to the subsumption tree. Note that we only add necessary concepts required for Algorithm 4.1. Therefore, the first property only takes $V_C \cup V_{NC}$ into a consideration, i.e., we are not able to assume $\prec$ relation between nodes in $V_C \cup V_{NC}$ and nodes in V_F.

Algorithm B1 to generate $\mathcal{G} = (V_C \cup V_{NC} \cup V_F,\ A)$:

---

1 Initially set $X = V_C$

2 For each concept name $v \in V_C$:

3 create node $c = \neg v$

4 compute set $X \downarrow c = \{x \in X | \, c \leq x$ and $c \not\leq y$ for all $y \prec x, y \in X\}$

5 compute set $X \uparrow c = \{x \in X | \, x \leq c$ and $y \not\leq c$ for all $x \prec y, y \in X\}$

6 add arcs between c and each element of $X \downarrow c$, and between c and each element of $X \uparrow c$

7 remove all arcs between elements of $X \uparrow c$ and $X \downarrow c$

8 compute set $X^{V_C} \parallel c = \{x \in X, \boldsymbol{x} \in V_C \mid \bot \leq c \sqcap x \}$

9 For each element $d \in X^{V_C} \parallel c$:

10 create artificial node $a_d = c \sqcap d$ and add an arc $(a_d, \bot)$ connecting $\bot$ to a_d

11 add an arc (d, a_d) connecting a_d to d and (c, a_d) connecting a_d to c

12 $X = X \cup \{c\}$

To extract the evidence set, Algorithm 4.1 computes Ω's in order to ensure the second requirement in Definition 4.1. Any set $\Omega_{m,P}$ depends on an overlap between two set of nodes corresponding to $C(o)$ and $\neg P_m$. To verify the overlap of these two sets, both V_{NC} and V_F are necessary. Algorithm B1 adds V_{NC} and V_F to the standard subsumption tree which only includes concept names V_C. It relies mainly on an incremental method involving the top and bottom search computed in steps 4 and 5. As arcs in the tree only represent the subsumption relationship, steps 8-11 further take into account an overlap case which involves nodes in V_F.

B.2 An efficient version of Algorithm 4.1 using a bisection method

As a more efficient version of Algorithm 4.1, we propose Algorithm B2 using a bisection method based on $\Omega_{m,P}$ as of follows.

Algorithm B2 $(o,\ \bar{O})$ with $\mathcal{G} = (V_C \cup V_{NC} \cup V_F,\ A)$:

---

1 Set $sup_{C(o)} = \emptyset$

2 For each $\bar{o} \in \bar{O}$:

3 Set $sup_{C(\bar{o})} = \emptyset$

4 For each $path\ P \in Path_{C(\bar{o})^\top}$:

5 $m = \left\lfloor \frac{length(P)}{2} \right\rfloor$

6 While True:

7 $\Omega_{m,P} = \{y \in V_C \cup V_{NC} |\ y \in Node_{\perp C(o)} \cap Node_{\perp \neg P_m},\ y \neq \perp\}$

8 If $\Omega_{m,P} \neq \emptyset$:

9 $\Omega_{m-1,P} = \{y \in V_C \cup V_{NC} |\ y \in Node_{\perp C(o)} \cap Node_{\perp \neg P_{m-1}},\ y \neq \perp\}$

10 If $\Omega_{m-1,P} = \emptyset$:

11 For m' = m to $length(P)$:

12 Add P_m, to $sup_{C(\bar{o})}$

13 Break

14 Else:

15 $m = \left\lfloor \frac{m}{2} \right\rfloor$

16 Else:

17 $m = m + \left\lfloor \frac{m}{2} \right\rfloor$

18 $sup_{C(o)} = sup_{C(o)} \cup sup_{C(\bar{o})}$

19 Remove duplicate nodes in $sup_{C(o)}$

20 Return α = SetCover($sup_{C(o)}$)

Note that as in Algorithm 4.1 $\forall a \in sup_{C(o)}$ there is $\bar{o}$ such that $P \in Path_{C(\bar{o})^\top}$ contains a due to steps 2-18. For any $\bar{o}_i \in \bar{O}$, Algorithm B2 identifies the m^{th} position in each path $P \in$

$Path_{C(\bar{o}_i)^\top}$ such that $\Omega_{m,P} \neq \emptyset$ and $\Omega_{m-1,P} = \emptyset$ by the bisection search method. We know that $P_m \sqsubseteq P_{m-1} \Leftrightarrow \neg P_{m-1} \sqsubseteq \neg P_m$ for any concepts P_m and P_{m-1}. Hence, $\Omega_{m-1,P} \neq \emptyset$ implies that $\Omega_{m,P} \neq \emptyset$. Step 12 adds all nodes which are subsumed by P_m to $sup_{C(\bar{o}_i)}$ if conditions in steps 8 and 10 are satisfied

As in the run time analysis of Algorithm 4.1, we let N be the number of nodes in $\mathcal{G}$ and M the maximum number of paths between any node and the root $\top$. Similarly to Algorithm 4.1, the most outer loop in step 2 is O(N) while the middle loop in step 4 is considered as O(M). Algorithm B1 relies on the bisection search which is accounted for O($\log_2 N$). Computing $\Omega_{m,P}$ for each node P_m requires O(N^2). Therefore, the computational complexity of Algorithm B1 is O($\log_2 N \cdot N^3 \cdot M$) compared to O($N^4 \cdot M$) corresponding to Algorithm 4.1.

B.3 Proof for proposition 4.1 and 4.2

Proof for Proposition 4.1:

We first argue that SC is feasible. For any path $P \in Path_{C(\bar{o}_i)^\top}$ of $\bar{o}_i \in \bar{O}$, $P_{length(P)} = C(\bar{o}_i) \in V_C$. According to the assumption that $\exists y \in V_C \cup V_{NC}, y \neq \perp$ such that $y \sqsubseteq C(o) \sqcap \neg C(\bar{o})$ for all $\bar{o} \in \bar{O}$ and o in KB, there exists node y in $\mathcal{G}$ which yields paths from $\neg C(\bar{o}_i)$ to y and $C(\mathrm{o})$ to y for any $\bar{o}_i \in \bar{O}$. This implies that $Node_{\perp C(o)} \cap Node_{\perp \neg C(\bar{o}_i)} \neq \emptyset$ which corresponds to nonempty $\Omega_{m,P}$ of $C(\bar{o}_i)$ for any $\bar{o}_i \in \bar{O}$ and any path P for an m. Hence, $T_{C(\bar{o}_i)}$ is a set in SC for every $\bar{o}_i \in \bar{O}$. This implies that SC is feasible.

Let $T_{\alpha_1}, T_{\alpha_2}, \ldots, T_{\alpha_k}$ be a feasible solution to SC. We know that $\bigcup_i T_{\alpha_i} = U$ and $\alpha_i \in sup_{C(o)}$. For each $\bar{o}_i$, there exists $j(i)$ such that $\bar{o}_i \in T_{\alpha_{j(i)}}$. Let $\alpha = \{\alpha_{j(i)}\}_{i=1}^{n}$. We next argue that α is an object evidence of falseness. The definition implies that $\alpha_{j(i)} \in P$ where $P \in Path_{C(\bar{o}_i)^\top}$ which implies that $C(\bar{o}_i) \sqsubseteq \alpha_{j(i)}$. The first requirement in Definition 4.1 is therefore satisfied.

We next show the second property in Definition 4.1. Each $\alpha_{j(i)} \in sup_{C(o)}$ which is added in step 9 has to satisfy conditions in steps 6 and 8 according to Algorithm 4.1. For any path $P \in Path_{C(\bar{o}_i)^\top}, \Omega_{\alpha_{j(i)},P} \neq \emptyset$ implies that $\exists y \in V_C \cup V_{NC}$, $y \neq \bot$ such that there is a path from $C(o)$ to y and a path from $\neg\alpha_{j(i)}$ to y. Hence, $\exists y \in V_C \cup V_{NC}$, $y \neq \bot$ such that $y \sqsubseteq C(o)$ and $y \sqsubseteq \neg\alpha_{j(i)}$ which is equivalent to $C(o) \not\sqsubseteq \alpha_{j(i)}$. As $C(o) \not\sqsubseteq \alpha_{j(i)}$ is assured for all $j(i)$, the second requirement in Definition 4.1 is satisfied. It is clear by construction that $k \geq |\alpha|$.

Proof for Proposition 4.2: Let $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ be feasible to EP.

By requirement 1 of EP, for every $\bar{o}_i \in \bar{O}$ there exists α_i such that $C(\bar{o}_i) \sqsubseteq \alpha_i$. From TBox classification which is used to construct the subsumption tree, we know that $a \sqsubseteq b$ if and only if there is a path from b to a for $a, b \in V_C$. Hence, a path $P \in Path_{C(\bar{o}_i)^\top}$ is one of the paths in step 4. The requirement 2 of EP corresponds to Ω' computed and substituted to step 7 in Algorithm 4.1. If $\Omega'_{m,P} \neq \emptyset$ which is verified (by checking satisfiability of the concept $C(o) \sqcap \neg P_m$) in Step 8, P_m considered as evidence α_i is added to $sup_{C(\bar{o}_i)}$ in step 9. By step 12 in Algorithm 4.1, $\alpha_i \in sup_{C(\bar{o}_i)}$ and thus $\alpha_i \in sup_{C(o)}$. Hence, there exists T_{α_i} for each $\alpha_i \in sup_{C(o)}$. Since $\alpha_i \in P$ where $P \in Path_{C(\bar{o}_i)^\top}$, $C(\bar{o}_i) \in T_{\alpha_i}$. We have $\bigcup_i T_{\alpha_i} \supseteq \bigcup_i C(\bar{o}_i) = \bar{O} = U$. This implies that α is

feasible to SC. Since a feasible solution to EP yields a feasible solution to SC, it implies that $Z_{SC} \leq Z_{EP}$. As $Z_{SC} \leq Z_{EP}$ and $Z_{SC} \geq Z_{EP}$ (from Proposition 4.1), $Z_{SC} = Z_{EP}$ is directly followed.

B.4 Examples of triplets in the KG

We manually observe extracted triplets in the KG and compare with their original statements. Based on our manual observation, incorrectly extracted triplets are provided in Table B.1.

Table B.1: Examples of incorrectly extracted triplets in the KG based on SemRep

Original statements	Extracted triplets
Six additional imino derivatives of pyridoxal have been studied, but none of these new compounds was as effective as PIH.	(Pyridoxal ; same as ; Prolactin Release-Inhibiting Hormone | PEE1)
There was no significant different in the levels of G6PD activity in subjects with GdA or GdB.	(Glucose-6-phosphate dehydrogenase measurement, quantitative ; USES ; GDA)
Two methods for the removal of erythrocytes from buffy coats for the production of human leukocyte interferon.	**(**Erythrocytes ; PRODUCES ; human leukocyte interferon)

B.5 Results from an experiment based on relation extraction algorithms

According to the experiment, we observe matches among KG, Ollie and LSTM-ER models. Table B.2 illustrates number of matches among different models.

Table B.2: Number of matches among KG, Ollie and LSTM-ER models based on 10,000 triplets

Number of triplets	KG	Ollie	LSTM-ER
160	×	×	×
160		×	×
525	×	×	
1500	×		×

B.6 Results from the PRA model based on lay triplets

We first identify the truthfulness of a lay triplet based on the PRA model for each relation type. A proportion of false triplets which is equivalent to number of false triplets divided by total number of triplets is computed. Object candidates are computed for false triplets only. Total number of triplets, a false triplet proportion and average number of object candidates for each relation types are summarized in Table B.3.

Table B.3: Statistics of truthfulness and object candidates obtained from the PRA model

Relation type	**Number of triplets**	**false triplets proportion**	$\boldsymbol{Avg\ |\bar{O}|}$
LOCATION_OF	378	0.15	12.19
ISA	319	0.07	10.43
PREDISPOSES	292	0.33	8.68
TREATS	207	0.26	10.80
CAUSES	179	0.39	11.50
AFFECTS	116	0.41	14.69
COEXISTS_WITH	114	0.30	17.29
PREVENTS	110	0.25	7.86
PART_OF	69	0.25	7.65
INTERACTS_WITH	46	0.30	3.79
INHIBITS	45	0.20	7.67
ASSOCIATED_WITH	37	0.22	8.63

AUGMENTS	35	0.46	11.75
USES	35	0.09	7.67
PRODUCES	30	0.17	8.20
DIAGNOSES	22	0.14	14.00
DISRUPTS	21	0.48	9.70
PRECEDES	16	0.25	17.25
METHOD_OF	13	0.38	19.2

B.7 Evidence sets based on a manual observation

We compare the cardinality of candidates, the cardinality of evidence, and all elements in the set corresponding to “Al” and “Ma” in Table B.4.

Table B.4: A complete comparison of elements in evidence sets “Al” and “Ma”

False Triplets	**Al**			**Ma**		
	$\|\bar{O}\|$	$\|\alpha\|$	α	$\|\bar{O}\|$	$\|\alpha\|$	α
Heparin TREATS **Fever**	2	1	'hemic system symptom'	3	1	'hemic system symptom'
Amiodarone TREATS **Hepatitis C**	5	1	'disease of anatomical entity'	9	1	'disease of anatomical entity'
Stress management TREATS **Mitral Valve Prolapse**	2	1	'nervous system disease'	2	1	'nervous system disease'
Capoten TREATS **Coughing**	12	1	'Disease, Disorder or Finding'	9	1	'Disease, Disorder or Finding'
Losartan TREATS **Varicose Ulcer**	5	3	'insulin resistance', ‘hypertrophy', ‘disease'	9	2	'ischemia', ‘disease'
Echocardiography DIAGNOSES **Hyperlipidemia**	4	1	'disease of anatomical entity'	4	1	'disease of anatomical entity'

Platelet Size DIAGNOSES **Anemia**	4	1	'Cardiovascular Diseases'	5	2	'Blood Platelet Disorders', ‘Cardiovascular Diseases'
Esophageal pH Monitoring DIAGNOSES **Malignant breast neoplasm**	11	5	'Biological Process', ‘Finding', ‘Non-Neoplastic Disorder', ‘Neoplasm by Morphology', ‘Digestive System Disorder'	8	3	‘Finding', ‘Non-Neoplastic Disorder', ‘Neoplasm by Morphology'
Cholesterol measurement test DIANOSES **Malignant breast neoplasm**	15	5	'Biological Process', ‘Mouse Disorder by Site', ‘Finding', ‘Non-Neoplastic Disorder', ‘Dependence'	4	2	'Non-Neoplastic Disorder by Site', ‘Finding'
Electrocardiogram DIAGNOSES **Muscle strain**	1	1	'cardiac disorder AE'	1	1	'cardiac disorder AE'
Caffeine CAUSES **Gout**	2	1	'Cell Physiological Phenomena'	8	3	'Phenomena and Processes Category', ‘Behavior and Behavior Mechanisms', ‘Signs and Symptoms, Digestive'
hypercholesterolemia CAUSES **Neuropathy**	7	2	'Finding', ‘Non-Neoplastic Disorder'	6	2	'Finding', ‘Non-Neoplastic Disorder'
Leukemia CAUSES **Gout**	3	2	'genetic disease', ‘neoplasm (disease)'	2	2	'genetic disease', ‘neoplasm (disease)'
Harpin CAUSES **Cardiomegaly**	7	2	'Pathologic Processes', ‘Phenomena and Processes Category'	2	1	'Cell Death'

Ascorbic Acid CAUSES **Senile Plaques**	2	2	'Atherosclerosis', 'Abnormality of the cerebral ventricles'	4	2	'Abnormality of digestive system physiology', 'Abnormality of nervous system physiology'