공학박사학위논문

Building Named Entity Knowledge Graph Using Named Entity Normalization

고유명사 정규화 기법을 이용한 지식 그래프 구축

2023 년 2 월

서울대학교 대학원

산업공학과

전 성 환

Building Named Entity Knowledge Graph Using Named Entity Normalization

고유명사 정규화 기법을 이용한 지식 그래프 구축

지도교수 조 성 준

이 논문을 공학박사 학위논문으로 제출함

2022 년 1 월

서울대학교 대학원

산업공학과

전 성 환

전성환의 공학박사 학위논문을 인준함

2022 년 1 월

위 원 장 _____이 재 욱_____ (인)

부위원장 _____조 성 준_____ (인)

위 원 _____백 복 현_____ (인)

위 원 _____이 영 훈_____ (인)

위 원 _____고 태 훈_____ (인)

Abstract

Building Named Entity Knowledge Graph Using Named Entity Normalization

Sung Hwan Jeon
Department of Industrial Engineering
The Graduate School
Seoul National University

Text mining aims to extract the information from documents to derive valuable insights. The knowledge graph provides richer information from various documents. Past literature responded for such needs by building technology trees or concept network from the bibliographic information of the documents, or by relying on text mining techniques in order to extract keywords and/or phrases. In this paper, we propose a framework for building a knowledge graph using named entities. The knowledge graph construction framework in this paper satisfies the following conditions: (1) extracting the named entity in the completed form, (2) Building datasets that can be trained and be evaluated by the named entity normalization models in various domains such as finance and technical documents in addition to bioinformatics, where existing NEN research has been active, (3) creating the better performing named entity normalization model, and (4) constructing the knowledge graph by grouping named entities with the same meaning that appear in various forms.

Keywords: Named entity normalization, Edge weight updating neural network, Text mining in bioinformatics, Text mining in finance, Text Mining in patent documents, Named entity graph, Knowledge graph, Patent graph, Keyword extraction

Student Number: 2016-21122

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The text mining technology is undergoing a rapid evolution thanks to the exponential growth in the number of text-rich documents available online, and as a result, it is being widely applied in a range of domains' documents such as finance documents, patent documents and bioinformatics documents. To organize and to derive valuable insights from the documents, building the knowledge graphs from the documents is the one of the most effective techniques. In this research, we propose named entity knowledge graph construction framework. To overcome drawbacks of previous knowledge graph construction models, we present the dictionary construction for named entity normalization, named entity normalization model using edge weight updating neural network, and building knowledge graph using named entity recognition and normalization models.

Documents subject to analysis contain many named entities, which are proper names that denote unique objects such as organizations, products, persons, and locations. The technique used to extract named entities from documents is called named entity recognition (NER, henceforth). Furthermore, named entity normalization (NEN, henceforth) involves matching extracted named entities with homogeneous identity and is pivotal for text mining tasks.

Identifying the matched named entity pairs is critical in financial text mining tasks. More precise named entity normalization in text mining will benefit other subsequent text analytic applications. Yet, there are insufficient public datasets for financial named entity normalization. We built the named entity normalization dataset from publicly open financial documents and patent documents using parsed named entity tokens with state of the art named entity recognizing model and concatenating named entity tokens with more precise token linking methodology. Our dataset contains major challenges in named entity normalization are (1) distinguishing synonyms, (2) linking abbreviations, (3) identifying acronyms, (4) recognizing different mixture of punctuations and alphabets, (5) matching entity with descriptive phrases and (6) correcting possible NER parsing errors.

The early NEN models explored knowledge-based approaches. Generating the rules for named entity matching based on domain knowledge is valid only for the dataset in which the corresponding rules are already created. The rule-based models are not robust for the neologisms. In order to overcome the disadvantage that the rule-based model is not robust, models based on machine learning have been introduced. However, machine learning models are limited to specific fields such as bioinformatics NEN and chemical engineering NEN due to lack of NEN datasets in other domains. Our research aims to construct fully automated NEN model that can be applied to various other domains. To test our model's robustness on different domain, we also apply the NEN dataset in finance.

We built the named entity normalization model with a novel Edge Weight Updating Neural Network. Our proposed model when tested on four different datasets achieved state-of-the-art results. We, next, verify our model's performance on NCBI

Disease, BC5CDR Disease, and BC5CDR Chemical databases, which are widely used named entity normalization datasets in the bioinformatics field. We also tested our model with our own financial named entity normalization dataset to validate the efficacy for more general applications. Using the constructed dataset, we differentiate named entity pairs. Our model achieved the highest named entity normalization performances in terms of various evaluation metrics.

An automated named entity normalization model reduce the burden of hand-mined information extraction tasks. Clear linkage between entities with different forms, such as abbreviations and acronyms, aid in more accurate sentiment analysis. The named entity normalization model also benefits the creation of more comprehensible classifying and clustering documents. One of the primary contributions of our study are (1) constructing better performing NEN model using an Edge weight updating neural network and (2) applying our proposed model to bioinformatics NEN and financial NEN tasks.

Recent technological discoveries have well been reflected through the rapid growth in the patent filings. The speed and the volume these patents are being generated call for an automated process, based on machine learning techniques, for cost-effective and timely analysis. Past literature responded for such needs by building technology trees or concept network from the bibliographic information of the patent documents, or by relying on text mining techniques in order to extract keywords and/or phrases. While these approaches provide an intuitive glance into the technological hotspots or the key features of the select field, there still is room for improvement, especially in terms of recognizing the same entities appearing in different forms so as to properly interconnect closely related technological concepts. In this paper, we

propose to build a patent knowledge network using USPTO's patent filings for the semiconductor device sector by fine-tuning Huggingface's named entity recognition model with our novel edge weight updating neural network. Experiment results show that our proposed approach performs very competitively against the conventional keyword extraction models frequently employed in patent analysis, especially for the named entity normalization and the document retrieval tasks. We also show that our model is robust to the out-of-vocabulary problem by employing the fine-tuned BERT NER model.

Chapter 2

Literature review

2.1 Named entity normalization dataset

Among many related text mining applications, named entity normalization can be applied to various text mining researches and text mining practices. In preprocessing for applying text mining techniques to solve real world problems, NER and NEN models are preformed preemptively. ShARe/CLEF [84] is one of widely used NEN dataset for bioinformatics which the dataset is constructed of clinical notes. NCBI [16] dataset contains PubMed abstracts for disease name normalization. TAC2017ADR [13] aims to link identical drug labels. Genes, proteins, and bacteria name normalization datasets are also available from BC2GM [80], BioNLP09 [38], and BioNLP-OST19 [8]. In chemical engineering, SCAI [44] and IUPAC [43] are available for researches on chemical name matching. Similar to chemical names, Weston et al. [87] constructed and distribute the dataset for material engineering to normalize entities to a canonical form.

However, the NEN dataset for the financial domain is scant and there is a need for developing a dataset targeting the financial NEN. Many researchers have developed targeted datasets for more general NEN tasks in domains such as user comments, product description, and financial invoices. For example, in their study, Jijkoun et al. [33] used user comments from newspaper websites. Sun et al. [81] performed normalization of product entity names, for which the dataset was developed by the authors. The study conducted by Francis et al. [18] on financial invoices is the most relevant one to our study. However, Francis et al. focused on insurance, telecommunications, banking, and tax companies using the following entities: International Bank Account Number (IBAN) of the beneficiary, invoice number, invoice date, and due date [18]. The focus of our study is on more general financial entity normalization, which covers entities from all financial sectors. Previous studies using the datasets illustrated above used various machine learning and deep learning models.

2.2 Named entity normalization

Bioinformatics, chemical engineering, and materials science domain actively adopt cutting-edge deep learning frameworks for NEN tasks. According to Cho et al. [9], various products exist for recognizing and normalizing named entities in biomedical fields such as ProMiner [26] and MetaMap [4]. DNorm [47] and TaggerOne [48] also used machine learning models such as pairwise ranking scoring and semi-Markov models, respectively, for NEN processing. In genetic engineering, GenNorm [86] and GNAT [25] are used to normalize the gene names. ChemSpot [72] uses Conditional Random Field for NER and NEN tasks in chemical engineering. Weston et al. [87] developed MatScholar [87] python repository to perform general NLP tasks on material science texts, which includes entity normalization.

Applying machine learning algorithms in the financial domain is gaining increasing attention. One major branch is stock movement forecasting using various deep learning mechanisms [5, 11]. Thanks to the rapid developments of unstructured data processing techniques, researches on applying text mining techniques to the financial fields have increased in number. In their study, Gupta et al. [24] illustrated the trends for applying text mining in finance. Among many related text mining applications in finance, NEN can be applied to various financial researches and financial practices.

There are similarities between the string matching methodologies in various other fields and NEN researches. Sun et al. [81] proposed NEN for product names using a pre-constructed product entity linkage dictionary. In semantic string matching, Siamese Neural Networks are widely used [60, 71, 55]. Krivosheev et al. [45] used Siamese Graph Neural Network for company name normalization. We need to extend NEN on company names to NEN on a wide range of product names and legal entities. Siamese RNN model successfully apprehends the morphological similarity between strings [62]. Niu et al. [63] applied Attention mechanisms for medical concept normalization. Furthermore, the evolution of Transformer-based models capacitate the adoption pre-trained language models such as BERT [14] for entity linking problems [61].

The major development in recent NEN researches is as follows. Kang et al. [35] proposed rule-based NLP model for better disease name normalization. Ghiasvand et al. [22] used edit distance based model for disorder mention normalization tasks. D‘Souza et al. [17] proposed an early NEN model using a rule-based model, which requires comparatively more human input when generating the rules. The model is static and, thus, there is a possibility that new rules need to be created when applying the model to other datasets. Rahmani et al. [69] proposed random walk applied on the augmented graph to link similar entities in genealogical graphs. NEN models that use more advanced machine learning and deep learning techniques can be more effective. In order to overcome the disadvantage that the rule-based model is not robust, models based on machine learning have been introduced [36, 32, 83]. Leaman et al. [48] used semi-Markov model, Li et al. [52] used word-level CNN model, and Wirght and Dustin [88] and Phan et al. [66] models based on BiGRU and BiLSTM. However, BERT achieved state-of-the-art performance in many general text mining and natural language processing (NLP) challenges. Compared with the four models illustrated above, the most recent researches such as the BERT ranking model [32] and BioSyn [83] takes full advantage of the BERT model by training the model based on BERT embeddings. The BERT Ranking model [32] used ranking-based objective function and BioSyn [83] used Synonym Marginalization techniques as the objective function for training. Our proposed model optimizes BERT embedding vectors with named entity graph's edge weight updating neural network. Our proposed model successfully captures the ground truth linkage between named entity graphs, achieving the highest accuracies. Previous NEN researches focus mainly on the NEN dataset from a specific domain. To test the efficacy of our model in more general NEN tasks, we evaluate our model with NEN datasets from both the bioinformatics domain and financial domain.

Many NEN researches explore semi-supervised learning models. Our proposed model is motivated by one of the leading semi-supervised models on images, Edge-Labeling Graph Neural Network for Few-shot Learning [37] (EGNN). The major difference between EGNN and our model is that EGNN labels an edge for each round of training but our model updates edge weights for top K connected entities. By capturing more node and edge information simultaneously for each round of training, the proposed model shows better performance compared with other NEN models.

2.3 Knowledge graph construction

Text mining techniques and their applications have received remarkable and rapidly growing attention as a means to acquire useful information from corpora of various backgrounds and characteristics. Technology management fields have responded by actively utilizing text mining approaches to process and analyze professionally written technological reports and other technology-related documents [64]. One of the most prevailing examples includes text-mining-based patent analysis: to date, numerous studies have attempted to analyze the patent documents to investigate contemporary technological trends, assess technological capabilities, and/or analyze the commercial value of select technologies [10]. Kim et al. [40], for instance, built a semantic network to analyze the "ubiquitous computing technology" by merging pre-determined keywords, recommended by experts in the field, from the patent claims. Patent claims were queried based on those pre-determined keywords, and the returned documents were characterized further by employing the k-means clustering algorithm. It is, however, very costly to pre-define manually the target technology-related keywords as the authors did in their study because it requires a great amount of background knowledge, time, and human labor during the process.

The number of studies on knowledge graph construction has grown rapidly. The importance of knowledge graphs is emphasized in [31, 21, 27]. Relatively earlier knowledge representations through the ontological graph and semantic web approaches for manufacturing are listed in [70]. Rahmani et al. [67, 68] proposed human disease network and human drug network based on protein-protein interaction. DDREL [2] is more recent research on constructing the drug-drug relation graph. Li et al. [54] constructed the knowledge graph from electronic medical records(EMRs). EMR2vec [15] suggested the platform which incorporated patient data and clinical trials by a medical ontological graph. Bipartite graph [90] and hypergraph [85] are also used to represent knowledge in graph form. Technology topic network which was built based on the patent documents aided to establish the improved R&D plannings [78]. Liu et al. [56] constructed the industrial knowledge graph based on various industrial documents and applied the knowledge graph to few-shot text classification problems. Similarly, kim et al. [39] and Sun et al. [82] proposed information retrieval technique using knowledge graphs.

Recently, the availability of NLP tools has led to the introduction of a wide range of automatic keyword extraction models. TechNet [75] is the leading example of such efforts, which was derived by applying word embedding algorithms to a massive amount of patent filings to establish the semantic relations between the technological terms presented as vectors on the same linear space. While these studies suggest meaningful approaches for extracting insights from patent filings, they still suffer from several limitations, such as entity matching and normalization. For example, the terms “CNN” and “convolutional neural networks” convey virtually the same meaning; yet, the standard word embedding approaches would vectorize these terms separately as independent entities. In this study, we attempt to address such issues by normalizing the named entities whose definitions are supposed to be aligned as identical by exploiting the edge updating neural network of our novel design, with triplet loss, as first proposed by [30].

Chapter 3

Dictionary construction for named entity normalization

3.1 Background

By constructing and distributing the NEN dataset for finance and patent documents will foster researches on general text mining in various fields. We construct the dataset for the financial NEN task from the annual reports (Form 10-K) of Standard and Poor's 500 listed companies. We aim to build the dataset that fulfills the need for financial NEN; the dataset includes (1) synonyms, (2) abbreviations, (3) acronyms, (4) different combinations of punctuations and alphabets, (5) descriptive phrases, and (6) possible NER parsing errors. A detailed explanation of primary data sources, data preprocessing steps, and dataset construction procedures are as follows. Fig. 3.1 demonstrates the overall flow diagram for NEN dataset construction.

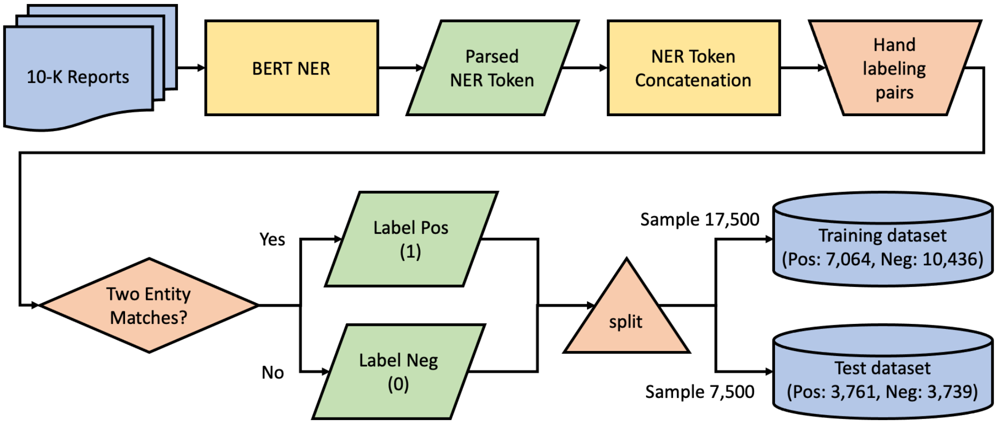


Figure 3.1: Flow diagram of the overall dataset construction

3.2 Dictionary construction methods

3.2.1 Finance named entity normalization dataset

Data source

We gather the year 2019's Form 10-Ks (published early 2020) of S&P500 companies from the U.S. firms and Exchange Commission (SEC) website[1], which is open to the public. We parse the business section of each 10-K documents from 496 companies. The business section of 10-K is considered the self-identity of firms and presents the information of main products, competitors, partners, and laws affecting the business. Among the sections in 10-K, this section contains the most number of entities. Out of 496 companies' business section, 67,792 sentences were parsed.

[1] https://www.sec.gov/edgar.shtml

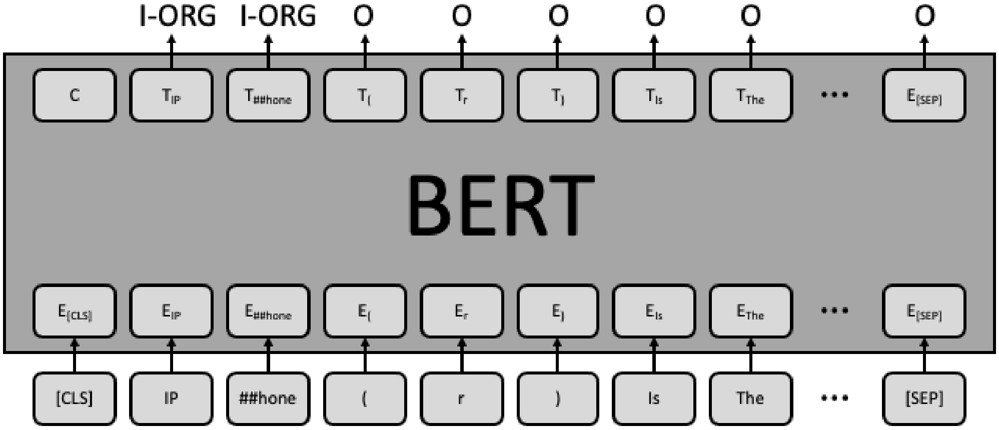


Figure 3.2: BERT named entity recognition in finance documents example

Data preprocessing

For NER in financial documents, we implement the BERT NER model [14] using Huggingface's[2] Python repository. Huggingface's NER model is trained using CoNLL-2003 NER dataset [74]. The outputs of the BERT NER model are WordPiece tokens that we have to link together with specified rules that will be circumstantially described below. There are four types of entity types: persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC), and one outside the named entity tag (O) in the CoNLL-2003 dataset. We detect entities with ORG and MISC tags.

Fig. 3.4 depicts named entity recognizing using pretrained BERT model for example sentence, "IPhone (r) Is the Company's line of smartphones based on its iOS operating system.", from Apple Inc's 10-K. The tag "IP" and "##hone" are successfully recognized and easily be concatenated. In addition, detecting "(r)" or "(tm)" mark (registered sign and trademark respectively) is very beneficial for fi-

[2] https://huggingface.co

nancial tasks. Detecting named entities with registered sign and trademark can be useful in distinguishing product named entities. Therefore, we decide to append such marks.

Table 3.1: Example of named entity token concatenation

Named Entity	Tokens	Token Types
Snapchat	S, ##nap, ##cha, ##t	ORG,ORG,ORG,ORG
Amazon.com, Inc	Amazon, ., com, ,, Inc	ORG,ORG,ORG,O,ORG
Anti-Bribery Laws	Anti, -, B, ##ri, ##bery, Laws	O,O,MISC,O,O,MISC
Boeing 737 (B737)	Boeing, 737, (, B, ##7, ##37,)	MISC,MISC,O,MISC,O,O,O
Google's Play Store	Google, ', s, Play, Store	ORG,O,O,ORG,ORG

We construct rigorous rule-based token concatenation model to detect named entity as comprehensive and interpretable as possible. Examples in Table 3.4 show how named entities are concatenated based on given BERT NER tokens.

- "Snapchat" is most basic and can be linked together with Huggingface's default NER concatenation package. If found named entity tag, "S", followed tokens are all tagged as named entity and starts with "##" which indicates that the token should be joined to previous token, all tokens should simply be connected.

- "Amazon.com, Inc" has "," token which is not labeld as named entity. However, "," links "Amazon.com" and "Inc" together. "Amazon.com" can be interpreted as website and comany name but "Amazon.com, Inc" clarifies that the entity is company. Therefore, we join tokens which follows similar patterns.

- "Anti-Bribery Laws" and "Bribery Las" should be distinguished. Without chaining "-" token, "Anti-Bribery Laws" can be separated, so concatenating the tokens around "-" token is needed.

- Named entity within the parenthesis are useful in detecting abbreviatons. “B737” and “Boeing 737” are two entities with same connotation. Entity “Boeing 737 (B737)” can play important role in named entity normalization as the key for linking entities “B737” and “Boeing 737”. We preserve entities within parenthesis.
- There are number of entities with “’s”. Token “’s” indicates the firms’ products like the example provided, “Google’s Play Store”. Sometimes, firm contains the “’s” for their company name like “DICK’S Sporting Goods Inc.”. For both scenarios, “’s” as possessive case and “’s” in proper noun, our algorithm chains the tokens around “’s”.

With the rule presented above, we complete named entity recognition as preprocessing for creating the financial NEN dataset. For year 2019 S&P500 firms’ 10-K, we parse total of 41,593 named entities.

Financial named entity normalization dataset construction

Table 3.2: Example of financial named entity normalization dataset

	Named Entity	Matching Named Entity
Synonyms	Coca-Cola ®	Coca-Cola
	COVID-19 Pandemic	COVID-19
	iPhone 11 Pro Max	iPhone ®
Abbreviations	Baker Hughes Company	Baker Hughes Co.
	Comcast Corporation	Comcast Corp.
	Qualcomm Incorporated	Qualcomm Inc.
Acronyms	Amazon Web Services	AWS
	Bank of New York Mellon	BNY Mellon
	New York Stock Exchange	NYSE
Combinations of punctuations	Apple, Inc.	Apple Inc.
	Walmart U. S.	Walmart U. S
	Booz Allen & Hamilton	Booz Allen Hamilton
Descriptive Phrases	EY (formerly Ernst & Young)	Ernst and Young
	Securities Exchange Act of 1934	(the Exchange Act)
	Facebook (including Instagram)	Facebook ®
NER Parsing Errors	Disney Channel-the	Disney Channel
	Full Throttle ®-a	Full Throttle ®)
	Keystone-our	Keystone Foods

With named entities recognized illustrated in Section 3.2.1, we construct the financial named entity normalization dataset. As mentioned in Section 3.1, our focus is to build a NEN dataset to meet the need for general text mining in finance; the dataset includes (1) synonyms, (2) abbreviations, (3) acronyms, (4) different combinations of punctuations and alphabets, (5) descriptive phrases, and (6) possible NER parsing errors. We hand label a total of 7,155 unique named entities into 2,600 groups; with each group sharing the same identity. Table 3.2 shows three examples in our dataset for types of named entities that need to be normalized.

- Synonyms:

  There exist entities with the suffix "®" or "™". "Coca-Cola ®" and "Coca-

Cola" are the same entity. In addition, "COVID-19 Pandemic" and "COVID-19" should be linked. We generalize the product model numbers in which "iPhone 11 Pro Max" and "iPhone ®" are considered identical entities.

- Abbreviations:

  Most abbreviations occur for abridging "Company" to "Co.", "Corporation" to "Corp.", and "Incorporated" to "Inc.".

- Acronyms:

  Acronyms are one of the most challenging NEN tasks. There are multiple abbreviations that are included in financial documents. We avoided matching acronyms if there are multiple original entities can be assigned. For example, "Advanced Development Programs (ADP)" and "Automatic Data Processing, Inc. (ADP)" both share the same acronyms, "ADP", but these should not be linked together.

- Combinations of punctuations:

  The different combinations of punctuations problems can be solved using rule-based approaches. However, there are many entities with a combination of punctuations. ",", ".", and "&" are commonly found and used interchangeably.

- Descriptive phrases:

  In parsed named entity, an entity with descriptive phrases can be frequently found. With or without descriptive phrases, the root or the identified entity is invariable.

- NER parsing errors:

No NER models and entity concatenation models are perfect. If NER is conducted manually, there are possible human errors too. According to our dataset, one common error model makes is appending the following token after "-" token. NER parsing error correction is one of the important targets our NEN model aims to achieve.

Table 3.3: Statistics of the financial named entity normalization dataset

	Train	Development	Test	Total
# of Identical Entity Groups	1,710	800	90	2,600
# of Positive Pairs	4,598	2,466	3,761	10,825
# of Negative Pairs	7,902	2,534	3,739	14,175
# of Pairs Total	12,500	5,000	7,500	25,000

Hand-matched entity pairs are labeled positive. We also added negatively labeled pairs in which two entities have no relationship. A total of 25,000 pairs with 10,825 positive matching pairs and 14,175 negative pairs are created. We separate entity groups for a train set, development set, and test set in which there are no overlapping groups. This eliminates possible training bias, especially when training the model with entities' graph topology. Table 3.3 shows the statistics of our financial NEN dataset. For training the models, we use train and development set for training and test set for evaluation similar to bioinformatics datasets.

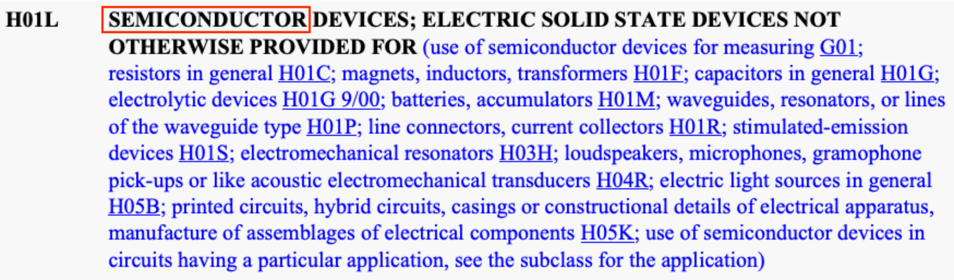
3.2.2 Patent named entity normalization dataset

Data source

In our experiment, we exploit USPTO data[3] from January, 2020 until the end of October, 2020. As our goal is to construct the patent knowledge graph from

[3] https://bulkdata.uspto.gov

semiconductor-related patent documents, we filter the patent claims by querying the word "semiconductor" in the description of the Cooperative Patent Classification (CPC), as shown in the example exhibited in Figure 3.3. The resulting dataset covers the following 12 CPC subclasses: H01C, H01F, H01G, H01L, H01M, H01P, H01R, H01S, H03H, H04R, H05B, and H05K. From the total of 35,734 documents, we recognized 69,812 named entities.



H01L **SEMICONDUCTOR DEVICES; ELECTRIC SOLID STATE DEVICES NOT OTHERWISE PROVIDED FOR** (use of semiconductor devices for measuring G01; resistors in general H01C; magnets, inductors, transformers H01F; capacitors in general H01G; electrolytic devices H01G 9/00; batteries, accumulators H01M; waveguides, resonators, or lines of the waveguide type H01P; line connectors, current collectors H01R; stimulated-emission devices H01S; electromechanical resonators H03H; loudspeakers, microphones, gramophone pick-ups or like acoustic electromechanical transducers H04R; electric light sources in general H05B; printed circuits, hybrid circuits, casings or constructional details of electrical apparatus, manufacture of assemblages of electrical components H05K; use of semiconductor devices in circuits having a particular application, see the subclass for the application)

Figure 3.3: Description of CPC subclass H01L

Semiconductor related patent named entity normalization dataset construction

For the NEN evaluation, we manually built a scarcely labeled dictionary matching named entities of different forms, yet with the same meanings. Out of 69,812 named entities, we hand-labeled 6,797 named entities to be matched with 1,000 unique named entity groups.

To extract named entities from the patent claims, we rely on the BERT NER model [14] provided by Huggingface's[4] Python package, in which the underlying model was pre-trained with the CoNLL-2003 NER dataset [74]. Four types of named entities are provided by the CoNLL-2003 dataset: persons (PER), organizations (ORG), locations (LOC), miscellaneous names (MISC), and those not recognizable

[4] https://huggingface.co

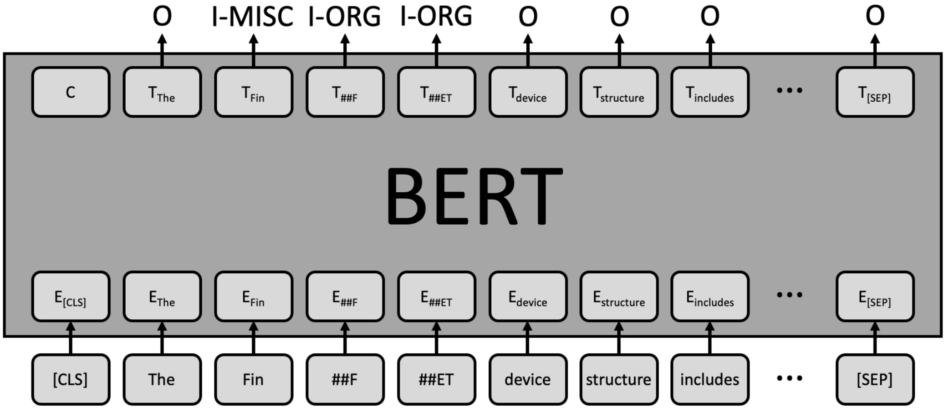


Figure 3.4: BERT named entity recognition in patent documents example

by the given dataset (O). Because our main focus is to extract technological concepts and terms, we detect entities labeled as ORG and MISC tags only.

Figure 3.4 depicts an example of the NER case using the pre-trained BERT model on the sentence “The FinFET device structure includes a second fin structure embedded in the isolation structure,” which actually appears in one of the semiconductor-related patent documents considered in our experiment.

The raw output of the BERT NER model is in the form of WordPiece [89] tokens, which are very difficult to interpret to the human eye at first glance. In this study, we enhance the human understanding of the preliminary NER results by conducting further token concatenation. More specifically, we construct a rigorous, rule-based token concatenation model to detect the named entities. Table 3.4 lays out the token concatenation scenarios we propose by the different concatenation types.

Table 3.4: Example of named entity token concatenation

Named Entity	Tokens	Token Types
FinFET	Fin, ##F, ##ET	MISC, ORG, ORG
Amazon.com, Inc	Amazon, ., com, ,, Inc	ORG, ORG, ORG, O, ORG
Anti-Hebbian	Anti, -, He, ##bb,##ian	MISC, O, MISC, MISC
Micro USB	Micro, USB	O, MISC
(NFC) tag	(, NFC,), tag	O, MISC, O ,O

The first case listed, where the tokens "Fin", "##F", and "##ET" are successfully recognized, is the simplest case that can be detected and easily concatenated utilizing Huggingface's default concatenation package. The term "FinFET" refers to one of the field effect transistors, and it will make sense only if the tokens "Fin" and the acronym "FET" are concatenated together. The tag of the token following "Fin" begins with "##", which indicates that it should be joined with the token appearing before itself. The following token, which ends with "##ET" suggests that the formerly merged term should be completed with the letters "F" and "ET", hence leading to the final form of the detected named entity as "finFET".

The next case is slightly more complicated yet easily solvable. Entities whose name includes punctuation marks such as periods (.) or commas (,), "Amazon.com, Inc", for example, require an extra step to be properly concatenated because these punctuation marks are recognized with the other (o) tags. In this case, we join the (o)-tagged token with the surrounding tokens if they are labeled with ORG tags.

Meanwhile, a compound noun, whose meaning changes owing to the combination with prefixes, such as "Anti-Hebbian" in the given example, should be distinguished from its original root word, "Hebbian". In this case, one needs to carefully concatenate the prefix "Anti-" with the following token "Hebbian" in order not to deteriorate the implication of the original wording.

The next row in the table presents the case where the major named entity token is decorated with a descriptive word or phrase, in this case “Micro USB”. We aim to keep as many information in the named entity as possible.

Our proposed token concatenator model binds such descriptive named entity tokens together effectively, hence providing richer understanding of the given corpus without less misleading results.

The last token concatenation scenario presents the case of the recognition and concatenation of tokens appearing in parentheses. For example, the pre-trained BERT NER model dissects the given token “(NFC) tag” into pieces; our proposed model, in contrast, preserves the parentheses intact with the acronym within as a unified entity. Thus, it provides the accurate interpretation that the detected entity is an acronym. Such instances are quite prevailing, especially in scientific documents, because acronyms appear rather frequently.

We hand-matched entities of the following six types: (1) synonyms, (2) abbreviations, (3) acronyms, (4) different combinations of punctuation and alphabets, (5) descriptive phrases, and (6) possible parsing errors. We report the examples of the matching named entities by type in Table 3.5.

Table 3.5: Matching categories and example for the named entity normalization dataset

	Named entity	Matching named entity
Synonyms	WiFi	IEEE 802 . 11
	Silicon Carbide .	SiC
Abbreviations	German Patent Appl .	German Patent Application
	Microsoft Corp .	Microsoft Corporation .
Acronyms	IoT	Internet of Things
	TFT	Thin Film Transistor
Combinations of punctuations	USB Type C	USB Type - C .
	Internet	“ Internet ” .
Descriptive phrase	Indium Tin Oxide	Indium Tin Oxide (ITO) ,
	LTE - A	LTE - Advanced (LTE - A) ,
Parsing errors	LEDs	Emitting Diodes (LEDs) can
	DRAM	Access Memory (DRAM) is

Table 3.6 presents an example of the resulting, manually built dictionary for semiconductor-related patent NEN.

Table 3.6: Excerpt from the semiconductor patent named entity normalization dataset

Named Entity	Group
Fin Field Effect Transistors (FinFETs) .	group_0
(FinFets) .	group_0
(e . g . , FinFETs) and	group_0
FinFet	group_0
WiFi .	group_29
WiFi 802 . 11	group_29
IEEE802 . 11 (WiFi) ,	group_29
802 . 11 (WiFi) ,	group_29
CD - ROMs (Compact Disc - Read Only Memories) ,	group_70
(CD - ROM) , Compact Disk	group_70
(CD) ROM	group_70
(CD - ROMs) , CD	group_70

We report the positive and negative entity pairs based on the matching status on the substring graph, as described in Section 5.2.1, in cross-check with our manually built dictionary, as mentioned in Section 3.2.2. Given the entity pairs connected on the substring graph, if the two entities are labeled to be in the same named entity group in our maunally built dictionary, the two entities are labeled positive. In contrast, if the two entities connected in the substring graph are placed in different groups, the two entities are labeled negative.

The detailed statistics are listed in Table 3.7.

Table 3.7: Statistics of the pairwise named entity matching evaluation dataset

	Number of positive pairs	Number of negative pairs	Total pairs
Train	25,695	29,241	54,936
Test	14,133	14,050	28,183
Total	39,828	39,745	83,119

Finally, we provide the basic summary statistics for the training and test sets in Table 3.8.

Table 3.8: Statistics of the semiconductor patent named entity normalization dataset

	Number of named Entity	Number of groups
Train	3,802	552
Test	2,995	448
Total	6,797	1,000

3.3 Chapter summary

In this chapter, we create financial NEN dataset using publicly opened financial documents and semiconductor-related NEN dataset using semiconductor-related patents. Our datasets covers six major NEN challenges: (1) synonyms, (2) abbreviations, (3) acronyms, (4) different mixture of punctuations and alphabets, (5) descriptive phrases and (6) possible NER parsing errors. Our proposed NEN datasets are used in Chapter 4 and Chapter 5.

Chapter 4

Named entity normalization model using edge weight updating neural network

4.1 Background

In the biomedical domain, disease names and chemicals in drugs often have different surface forms while sharing the same concept. Types of named entities with different surface forms that share same concept can be divided into following categories: (1) synonyms, (2) abbreviations, (3) acronyms, (4) different combinations of punctuations and alphabets, (5) descriptive phrases, and (6) possible NER parsing errors. For example, “hepatomegaly” and “liver enlarged” do not have matching strings but the two disease names have identical meanings, and thus, these two named entities are synonyms. Biomedical named entities have a wide variety of different surface forms compared with entities from other text sources. More accurate named entity normalization techniques will potentially improve the quality of downstream tasks. Moreover, matching entity pairs such as “International Business Machines” and “IBM”, which are examples of acronyms, are very critical in financial text mining applications. Linking entities with the same identity enables accurate sentiment analysis on firms and products. Furthermore, evaluation of news impacts on the stock market requires the connection between news articles and related firms. Given the wide range of named entities in bioinformatics and finance documents, the total number of tokens to be calculated for text clustering and classification is enormous.

The proposed method, that is, the edge weight updating neural network, consists of four parts: (1) ground truth entity graph construction, (2) similarity-based entity graph construction, (3) edge weight updating neural network training, and (4) edge weight updating neural network inferencing. The main concept behind the Edge Weight Updating Neural Network is to minimize the Ground Truth Entity Graph's edge weight distributions and the Similarity-Based Entity Graph's edge weight distributions. By minimizing the edge weight distributions on the two graphs, entity embeddings capture more accurate information on semantic similarity between matching entities.

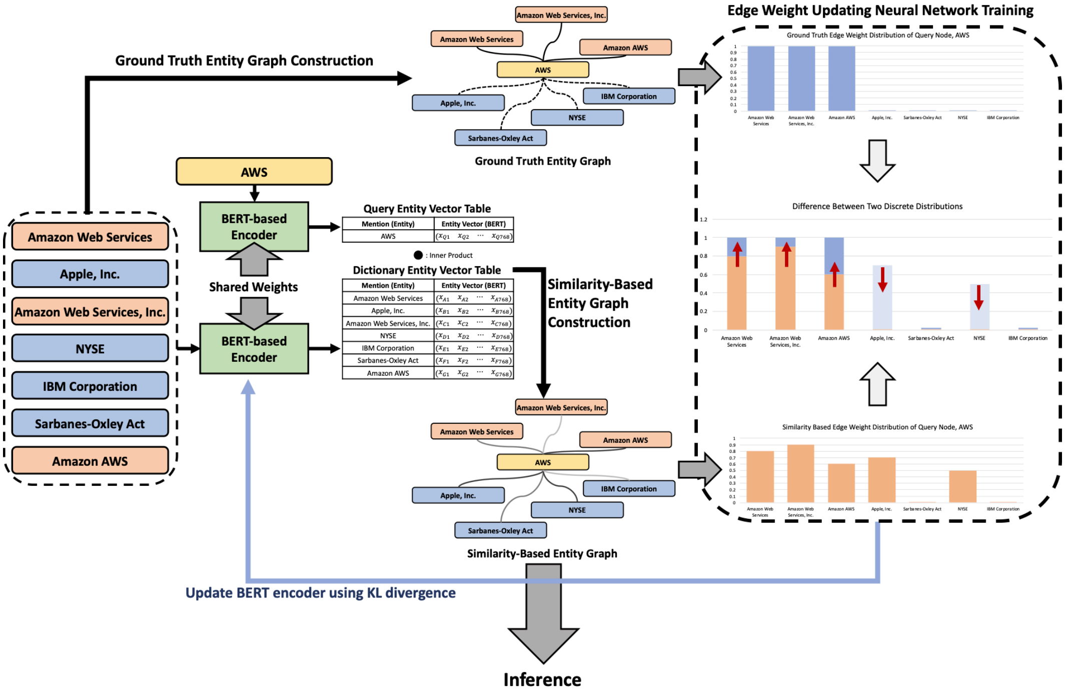
Our proposed model is evaluated on three widely used bioinformatics datasets (NCBI Disease, BC5CDR Disease, and BC5CDR Chemical) and its performance is compared with other cutting-edge models. Furthermore, to validate the efficacy of our proposed model in general NEN tasks, we construct a financial NEN dataset with state-of-the-art NER using BERT [14]. Using the constructed dataset, we propose the deep learning model to solve more practical financial NEN tasks. Out dataset incorporates major challenges in entity matching: (1) synonyms, (2) abbreviations, (3) acronyms, (4) different combinations of punctuations and alphabets, (5) descriptive phrases, and (6) possible NER parsing errors. Compare with other recent NEN models, our proposed model shows higher accuracies in all datasets used in the experiments, and our model is tested with not only bioinformatics NEN datasets but also financial NEN datasets, which verifies the efficacy in general NEN tasks.

4.2 Proposed model

Our proposed model, Edge Weight Updating Neural Network, consists of four major parts.

1. Ground Truth Entity Graph construction
2. Similarity-Based Entity Graph construction
3. Edge Weight Updating Neural Network training
4. Edge Weight Updating Neural Network inferencing

The basic idea behind Edge Weight Updating Neural Network is to minimize the Ground Truth Entity Graph's edge weight distributions and the Similarity-Based Entity Graph's edge weight distributions. The detailed flow diagram of Edge Weight Updating Neural Network is represented in Figure 4.1. Our motivation for constructing Edge Weight Updating Neural Network is to provide more positive and negative samples for training at once. Training the model to minimize the ground truth data and reconstructed data is more widely used in the deep learning models in computer vision such as GAN [23]. We apply these ideas to text mining and create the model that is trained to minimize the edge weight distributions of the ground truth entity graph and that of the similarity-based entity graph. In Figure 4.1, AWS is our query entity. First, with the vanilla BERT encoder, unrelated entities such as Apple, Inc. and NYSE might have higher similarity scores(edge weight between two entities) than the entity Amazon AWS. Then, the similarity-based graph is constructed with the given query entities according to the current similarity scores. The graph's edge weight distribution is compared with the ground truth entity graph's edge weight distributions. Entity embeddings are trained with Kullback-Leibler divergence [46] loss between two graphs. After iterating trough these steps, the BERT encoder will be trained to calculate the two entities' ground truth similarity score.



Query Entity Vector Table

Mention (Entity)	Entity Vector (BERT)
AWS	$(x_{Q1} \quad x_{Q2} \quad \cdots \quad x_{Q768})$

● : Inner Product

Dictionary Entity Vector Table

Mention (Entity)	Entity Vector (BERT)
Amazon Web Services	$(x_{A1} \quad x_{A2} \quad \cdots \quad x_{A768})$
Apple, Inc.	$(x_{B1} \quad x_{B2} \quad \cdots \quad x_{B768})$
Amazon Web Services, Inc.	$(x_{C1} \quad x_{C2} \quad \cdots \quad x_{C768})$
NYSE	$(x_{D1} \quad x_{D2} \quad \cdots \quad x_{D768})$
IBM Corporation	$(x_{E1} \quad x_{E2} \quad \cdots \quad x_{E768})$
Sarbanes-Oxley Act	$(x_{F1} \quad x_{F2} \quad \cdots \quad x_{F768})$
Amazon AWS	$(x_{G1} \quad x_{G2} \quad \cdots \quad x_{G768})$

Ground Truth Entity Graph Construction
Amazon Web Services, Inc.
Amazon Web Services
Amazon AWS
AWS
Apple, Inc.
IBM Corporation
NYSE
Sarbanes-Oxley Act
Ground Truth Entity Graph
AWS
BERT-based Encoder
Shared Weights
BERT-based Encoder
Amazon Web Services
Apple, Inc.
Amazon Web Services, Inc.
NYSE
IBM Corporation
Sarbanes-Oxley Act
Amazon AWS
Similarity-Based Entity Graph Construction
Similarity-Based Entity Graph
Update BERT encoder using KL divergence
Inference
Edge Weight Updating Neural Network Training
Ground Truth Edge Weight Distribution of Query Node, AWS
Difference Between Two Discrete Distributions
Similarity Based Edge Weight Distribution of Query Node, AWS
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.2
Amazon Web Services
Amazon Web Services, Inc.
Amazon AWS
Apple, Inc.
Sarbanes-Oxley Act
NYSE
IBM Corporation

Figure 4.1: Diagram of edge weight updating neural network with the query entity, AWS

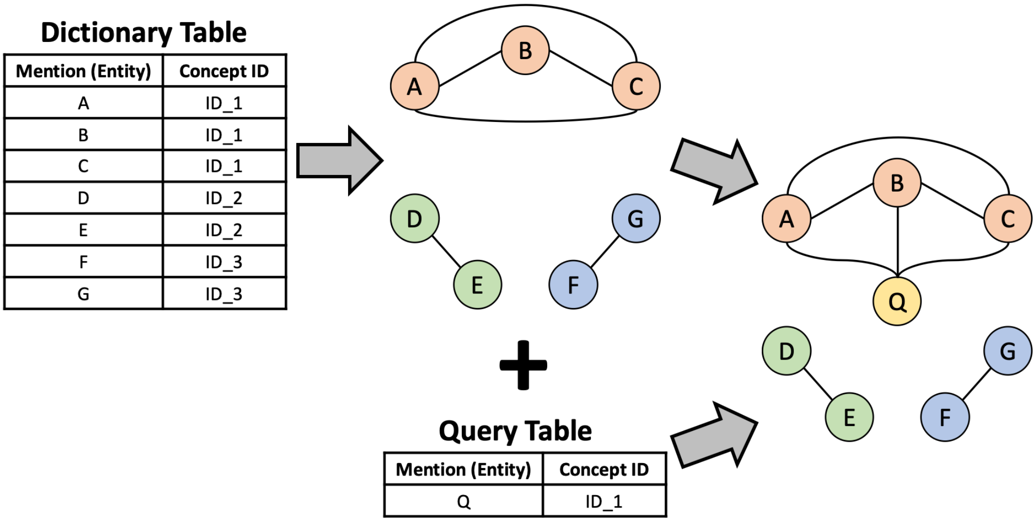
We use the BERT model for the named entity embeddings. There are two main reasons for choosing the BERT model for the named entity embeddings. There exist various pretrained BERT models that serve the specific purposes such as BioBERT [49] for bioinformatics documents, FinBERT [3] for finance documents, and PatentBERT [50] for patent documents. Compared to other language embedding models such as Word2Vec [58], Glove [65], and Fasttext [7], BERT model's WordPiece tokenizer is more robust handling the out of vocabulary problems. The number of out of vocabulary entities using the pretrained language models listed above are tabulated in Table 4.1.

Table 4.1: Number of out of vocabulary entities using pretrained language models

Model (Total # entities)	Word2Vec	Glove	Fasttext	BERT-based models
NCBI Disease (73,181)	8,469	7,991	7,279	0
BC5CDR Disease (73,548)	8,526	8,077	7,319	0
BC5CDR Chemical (407,428)	87,859	70,683	70,091	0
Financial NEN Dataset (24,195)	3,871	4,300	3,615	0

Detailed steps for constructing the Ground Truth Named Entity Graph, building the Similarity-Based Entity Graph, and training and inferencing the Edge Weight Updating Neural Network are presented in Sections 4.2.1, 4.2.2, 4.2.3 and 4.2.4, respectively.

4.2.1 Ground truth entity graph construction


Dictionary Table

Mention (Entity)	Concept ID
A	ID_1
B	ID_1
C	ID_1
D	ID_2
E	ID_2
F	ID_3
G	ID_3

Query Table

Mention (Entity)	Concept ID
Q	ID_1

Figure 4.2: Ground truth named entity graph construction

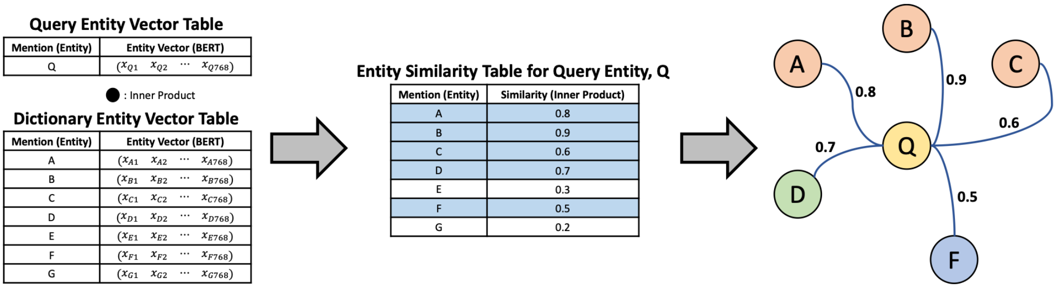
Ground Truth Entity Graph constructions are based on entity mentions in each dataset and their mapping concept IDs. Figure 4.2 demonstrates the steps for building the graph.

For the NEN corpus, each entity is annotated with one or more concept IDs. For example in Figure 4.2, entities A, B, and C share the same concept ID, ID_1. Then, entities A, B, and C are fully connected in the entity graph. Other entity pairs, D - E (concept ID: ID_2) and F - G (concept ID: ID_3) are linked. The training dataset for each NEN corpus has query entities with corresponding concept ID. If query entity Q has a concept ID of ID_1, then, query entity Q will be linked to entities

A, B, and C in the pre-constructed graph. As the constructed graph is the ground truth graph, each edge weight in the graph is 1.

We iterate all the entities in training sets that include the referencing dictionary entity table and the query entity table. Graph created by the following steps above is the Ground Truth Entity Graph which is the reference or the target graph the Similarity-Based Entity Graph will try to match.

4.2.2 Similarity-based entity graph construction



Query Entity Vector Table

Mention (Entity)	Entity Vector (BERT)
Q	$(x_{Q1} \quad x_{Q2} \quad \cdots \quad x_{Q768})$

● : Inner Product

Dictionary Entity Vector Table

Mention (Entity)	Entity Vector (BERT)
A	$(x_{A1} \quad x_{A2} \quad \cdots \quad x_{A768})$
B	$(x_{B1} \quad x_{B2} \quad \cdots \quad x_{B768})$
C	$(x_{C1} \quad x_{C2} \quad \cdots \quad x_{C768})$
D	$(x_{D1} \quad x_{D2} \quad \cdots \quad x_{D768})$
E	$(x_{E1} \quad x_{E2} \quad \cdots \quad x_{E768})$
F	$(x_{F1} \quad x_{F2} \quad \cdots \quad x_{F768})$
G	$(x_{G1} \quad x_{G2} \quad \cdots \quad x_{G768})$

Entity Similarity Table for Query Entity, Q

Mention (Entity)	Similarity (Inner Product)
A	0.8
B	0.9
C	0.6
D	0.7
E	0.3
F	0.5
G	0.2

Figure 4.3: Entity matching graph based on entity similarity construction

For each query entity, Similarity-Based Entity Graph is constructed as follows. Graph edges are calculated using BERT embedding vector similarities. We use

BioBERT [49] for bioinformatics NEN corpus' initial BERT embeddings and the original BERT [14] for financial NEN corpus' initial BERT embeddings.

For example, in Figure 4.3, let query entity Q has size of 768 (vector length of BERT embeddings), $Embed_Q = (X_{Q1} \quad X_{Q2} \quad \cdots \quad X_{Q768})$. Similarly, BERT-based entity embeddings in the dictionary set are also denoted as $Embed_{entity} = (X_{entity1} \quad X_{entity2} \quad \cdots \quad X_{entity768})$. The BERT embedding has a fixed length of 768, so our embedding vectors have a vector length of 768.

To calculate the edge weights based on entity similarities, we calculate inner products between query entities and dictionary entities. Since the calculation time for the matrix multiplication is fast, computing the similarity between every query entities and dictionary entities are relatively less time consuming. The largest dataset, BC5CDR Chemical, with the matrix size of (407,454 * 768) by (1,317 * 768) takes about 0.77 seconds on CPU and 0.27 seconds on GPU. The fastest calculation time on CPU is NCBI Diesease, (72,887 * 768) by (1,587 * 768), which take 0.20 seconds. The fastest calculation time on GPU is Finance NEN dataset, (20,071 * 768) by (20,071 * 768), which take 0.042 seconds. $< \; , \; >$ is the notation for inner product and Sim_Q is the set of similarities between query entity Q and all the entities in a dictionary; then the similarity between each query entity and each dictionary entity calculation is expressed as Equation 4.1,

$$Sim_Q = \{Sim \mid Sim =< Embed_Q, \; Embed_D > \; for \; D \in Dictionary\}$$

where,

$$Embed_{Q \in QueryEntities} = (X_{Q1} \quad X_{Q2} \quad \cdots \quad X_{Q768}),$$

$$Embed_{D \in Dictionary} = (X_{D1} \quad X_{D2} \quad \cdots \quad X_{D768}) \tag{4.1}$$

We normalize the similarity score by dividing the maximum similarity score in each query entity's similarity score set, Sim_Q. For Similarity-Based Entity Graph, top K edges based on similarity score are selected. Highlighted blue region in entity similarity table for query entity Q in Figure 4.3 demonstrates the edge weight determination steps when $top\_k = 5$. Mathematically, edge weights are calculated using Equation 4.2.

$$ConnectedEdgeWeight = \{Weight_{Edge} \mid Edge \in ConnectedEdge\}$$

where,

$$\begin{aligned}
&ConnectedEdge = argmax_{top\_k}\{Weight_Q\},\\
&Weight_Q = \left\{Weight \mid Weight = \frac{Sim}{Max_Q}\ for\ Sim \in Sim_Q\right\},\\
&Max_Q = max\{Sim_Q\}
\end{aligned} \tag{4.2}$$

For each training epoch, which is illustrated in Section 4.2.3, edge weights are updated. Updated entity embedding vectors generate new similarity scores that alter the edge weights in the graph.

4.2.3 Edge weight updating neural network training

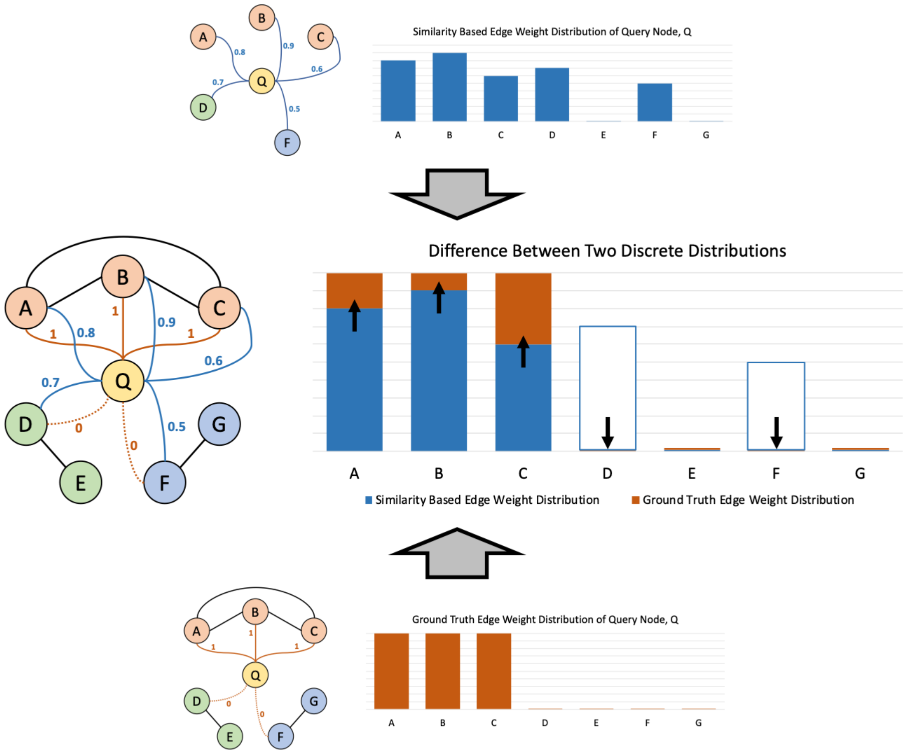

Similarity Based Edge Weight Distribution of Query Node, Q
A
B
C
D
E
F
G
Q
0.8
0.9
0.6
0.7
0.5
Difference Between Two Discrete Distributions
1
0
■ Similarity Based Edge Weight Distribution
■ Ground Truth Edge Weight Distribution
Ground Truth Edge Weight Distribution of Query Node, Q

Figure 4.4: Minimizing the edge weight distributions in edge weight updating neural network for query entity Q

The main concept of Edge Weight Updating Neural Network is to minimize the difference between the edge weights' discrete distribution for each query entity in the Ground Truth Entity Graph and the Similarity-Based Entity Graph. The Similarity-Based Entity Graph is dynamic. For the each iteration in the training phase, the Similarity-Based Entity Graph is reconstructed with the updated BERT model's parameters from the previous traning iteration. As illustrated in Section 4.2.2, edge weights are calculated by entities' embeddings. In each training epoch in Edge Weight Updating Neural Network, baseline BERT model's parameters are optimized to mimic the ground truth edge weight distributions.

Figure 4.4 shows the training process of our proposed model for the number of connected edges in the Similarity-Based Entity Graph is 5 ($top\_k = 5$). Following the example in Section 4.2.2, query entity Q is connected to dictionary entities A, B, C, D, and F, and edge weights are 0.8, 0.9, 0.6, 0.7, and 0.5, respectively. Given the Ground Truth Entity Graph in Section 4.2.1, the truth edge weights for connected edges between query entity Q and dictionary entities, A, B, C, D, and F are 1, 1, 1, 0, and 0, respectively.

In training procedures, BERT parameters are tuned to make edge weights distributions in Similarity-Based Entity Graph closer to the ground truth edge weight distributions. We use Kullback-Leibler Divergence Loss [46](KL divergence loss, henceforth) for training our model. As edge weight distribution is discrete, we normalize the edge weights using the Softmax function.

We denote graph as G, entity as V, and edge as E. The Ground Truth Entity Graph and the Similarity-Based Entity Graph are denoted as $G_{GT} = (V_{GT}, E_{GT})$ and $G_{Sim} = (V_{Sim}, E_{Sim})$, respectively. The adjacency matrices for Ground Truth Entity Graph and the Similarity-Based Entity Graph are denoted GT_A and Sim_A. $P_{Sim\_Edge_Q}$ is the discrete distribution of edge weights of Q in Similarity-Based Entity Graph. $P_{GT\_Edge_Q}$ is the discrete distribution of edge weights of Q in the Ground Truth Entity Graph. Our KL divergence loss is calculated using Equation 4.3.

$$Loss = KL(P_{GT\_Edge_Q} \mid\mid P_{Sim\_Edge_Q}) = P_{GT\_Edge_Q} \cdot \log \left(\frac{P_{GT\_Edge_Q}}{P_{Sim\_Edge_Q}} \right)$$

where,

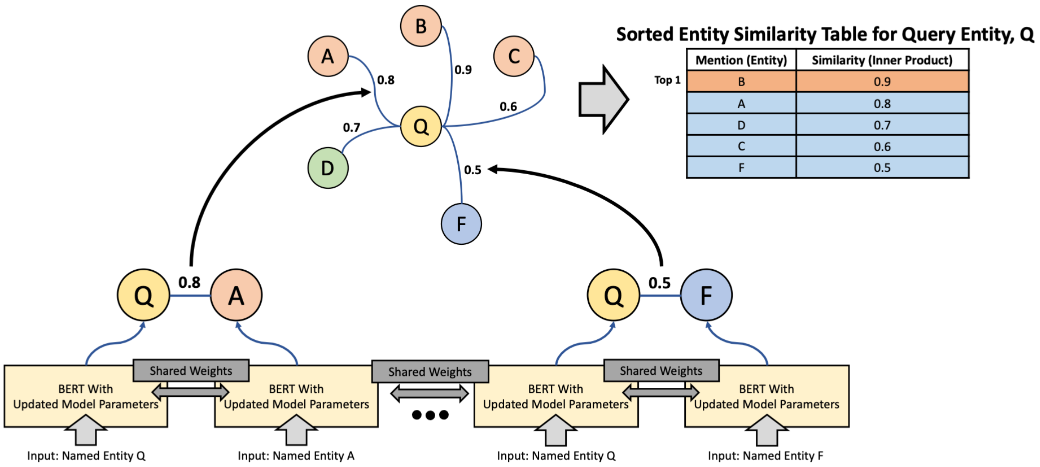
$$
\begin{aligned}
&P_{GT\_Edge_Q} = Softmax(GT\_Edge_Q),\\
&P_{Sim\_Edge_Q} = Softmax(Sim\_Edge_Q),\\
&GT\_Edge_Q = \{GT\_A_{query,d} \mid d \in argmax_{top\_k}\{Sim\_A_Q\}\},\\
&Sim\_Edge_Q = \{Sim\_A_{query,d} \mid d \in argmax_{top\_k}\{Sim\_A_Q\}\},
\end{aligned}
$$

and, A_Q is the edge weight vector connected to the given query entity node Q

(4.3)

We use an Adam optimizer with weight decay [57], and set the batch size to 16 and the number of connected edges in the Similarity-Based Entity Graph to 30 ($top\_k = 30$) for all datasets we test. We train our model for 50 epochs. The best scores are reported in Section 4.3.3.

4.2.4 Edge weight updating neural network inferencing



Sorted Entity Similarity Table for Query Entity, Q

Mention (Entity)	Similarity (Inner Product)
B	0.9
A	0.8
D	0.7
C	0.6
F	0.5

Figure 4.5: Inferencing the edge weight distributions in edge weight updating neural network for query entity Q

First, fine-tuned BERT embeddings illustrated in Section 4.2.3 are used to embed unseen query entities in test sets. With newly computed BERT embedding vectors, we repeat the steps in Section 4.2.2 to construct the new Similarity-Based Entity Graph. For each query entity, a dictionary entity with the highest edge weights is returned as a synonym. Figure 4.5 demonstrates the inferencing process of the Edge Weight Updating Neural Network.

4.3 Experiment results

4.3.1 Datasets

Most NEN researches are from the bioinformatics domain. To test our model's performance with other NEN models, we select three of the most used bioinforinmatics NEN datasets: NCBI Disease [16] and two datasets from Biocreinative V CDR (BC5CDR, henceforth) [53].

Three datasets summarized below contains bioinformatics-related entity mentions with unique concept IDs. The main goal of these datasets is to identify the mentions that share the same concept IDs. We follow NEN preprocessing convention for the datasets below, in which the mentions that do not exist in the concept dictionary are eliminated [66]. Bioinformatics NEN datasets usually consist of train, development, and test sets. Following previous studies, we use train and development sets for training our model. Test sets are used for evaluations.

Table 4.2: Data statistics of three bioinformatics NEN datasets

	# of Documents			# of Mentions (Entities)		
	Train	Dev	Test	Train	Dev	Test
NCBI Disease	592	100	100	5,134	787	960
BC5CDR Disease	500	500	500	4,182	4,244	4,424
BC5CDR Chemical	500	500	500	5,203	5,347	5,385

NCBI Disease [16]. NCBI Disease corpus provides disease mentions in different surface forms. Disease mentions in this dataset are extracted from 793 PubMed abstracts containing a total of 6,892 disease mentions, which are mapped to 790 unique disease concepts. Disease concepts are annotated by Medical Subject Headings (MeSH) and Online Mendelian Inheritance in Man (OMIM). Disease mentions sharing the same disease concept are considered synonyms. Table 4.2 shows detailed statistics of the NCBI Disease corpus.

Biocreative V CDR Disease and Biocreative V CDR Chemical [53]. The BC5CDR corpus is organized for challenging tasks of disease named entity recognition and chemical-induced disease relation extraction. The BC5CDR corpus consists of 1,500 PubMed articles with 4,409 annotated chemicals, and 5,818 disease and 3,116 chemical-disease interactions [53]. The dataset contains disease mention corpus and chemical mention corpus. Disease mentions are mapped into the MeSH IDs similar to the NCBI Disease corpus. Chemical mentions are annotated using the Comparative Toxicogenomics Database (CTD) [12]. Mentions that share the same disease concept and chemical concept based on MeSH ID and CTD ID are considered synonyms. Detailed statistics of both BC5CDR Disease corpus and BC5CDR Chemical corpus are illustrated in Table 4.2.

Financial named entity normalization dataset We use the dataset described in Section 3.2.1.

4.3.2 Experiment settings: named entity normalization in bioinformatics

We compare our proposed model's performance with seven different biomedical NEN models. The accuracy score presented in this study is excerpted from original papers. A summary of each model is illustrated in Table 4.3.

Table 4.3: Models used in bioinformatics NEN datasets evaluations

Models	Descriptions
Sieve-based [17]	This is one the earliest NEN papers. The research was conducted with 10 Sieve, which is mostly a rule-based approaches. Many published post this research follow similar preprocessing steps.
Taggerone [48]	Taggerone used the semi-Markov model for both NER and NEN tasks. Taggerone was originally validated on the NCBI Disease and BC5CDR corpus.
CNN Ranking [52]	CNN Ranking model used a word-level deep learning approach for NEN. This research did not perform better than the previous model, Taggerone. However, it was the first study that applied deep learning to NEN tasks.
NormCo [88]	NormCo used BiGRU, which is considered to be a better performing deep learning model with text data. NormCo achieved similar accuracy scores with significantly fewer parameters.
BNE [66]	BNE introduced two-level BiLSTM to capture both character-level and word-level information of biomedical entities, achieving increased NEN performance.
BERT Ranking [32]	BERT Ranking model is based on Transformer-based embeddings that use the pre-trained BERT [14], BioBERT [49], and ClinicalBERT [79] for their entity embeddings. For each entity, candidate concepts were retrieved and three different BERT models are fine-tuned to rank and to capture the ground truth concepts.
TripletNet [59]	The concept of TripletNet [29] for semi-supervised learning was introduced for NEN tasks. This study uses CNN for entity embedding and shared CNN parameters are trained with TripletNet structure.
BioSyn [83]	BioSyn uses BioBERT for entity embeddings and trained with Synonym Marginalization. Marginal Maximum Likelihood (MML) is the objective function for Synonym Marginalization.

4.3.3 Experiment Settings: Named Entity Normalization in Finance

The dataset we used is covered in Section 3.2.1. Table 4.4 shows each model used in NEN in Finance is tested. BioSyn is one of the state-of-the-art NEN model and the model's code is opened to public. We modified BioSyn for NEN dataset for finance domain and compared the performance. The experiments are conducted using Intel Core-i9-10940X CPU with 128GB memory and three NVIDIA GeForce Titan RTX GPU. To avoid possible biases caused by exogenous variables, we use the same setting for all models if applicable.

Table 4.4: Models used in finance NEN datasets evaluations

Models	Descriptions
Edit Distance [51]	Edit Distance is suitable for basic NEN tasks for linking “Apple Inc” and “Apple Inc.”. However, Edit Distance can only capture the superficial morphological similarity between two entities. In our experiment, we calculate the Edit Distance between two entity pairs and train a simple classifier to determine the equivalence of two entities.
BERT [14]	BERT is a state-of-the-art model for various NLP tasks. However, for our specific tasks, the BERT model has a limitation on capturing morphological similarity between entity pairs. We use pre-trained BERT vectors with size 768 and train a simple MLP classifier with batch size 4096 to determine the linkage between entity pairs.
Siamese GCN [42]	We use the entity graph illustrated in Section 4.2.2 and we use a pre-trained BERT vector for each entity node vector. 2-layer Siamese GCN is used in our experiment with 256 hidden nodes for the first GCN layer and 16 hidden nodes for the second GCN layer. GCN requires more epochs for training so we trained for 120 epochs for the full dataset (full batch: 17,500 entity pairs). The learning rate for ADAM optimizer for GCN is 0.01.
Siamese BiLSTM [77]	For Character Level Siamese BiLSTM model training, we one-hot encoded the characters entity strings with unique 85 tokens. We stack two BiLSTM layers. The BiLSTM cells in the first layer return 64 dimension hidden states output and the BiLSTM cells in the second layer return 16 dimension hidden states output. To prevent overfitting, we train the BiLSTM model for 12 epochs. The BiLSTM model is trained with a learning rate of 0.001. Embedding dimension, 16, is the same as GCN.
BioSyn [83]	The detailed model description is illustrated in Table 4.3.

For the pairwise NEN tasks, we take slightly different approach for BioSyn and our proposed model. The convention for the performance test in NEN tasks in bioinformatics is more similar to retrieval tasks. In bioinformatics NEN performance test, the model retrieve most similar entity from the candidate concepts (mentions). For each trial, the model scores if the retrieved named entity's concept ID matches the query entity's concept ID. However, for the pairwise NEN tasks, the model will retrieve the top-k most similar entities for both entities in the pair. We test BioSyn and our proposed model with the top-k of 1 and 3. The model scores if there exists the overlapping concept IDs in two groups of retrieved entities. For top-1, the model will recommend the one most similar entity and their concept IDs will be compared, and for top-3, the model will recommend the three most similar entities and will be scored by the presence of any overlapping concept IDs in two entity groups.

4.4 Results

We conduct both quantitative and qualitative analysis. For NCBI Disease, BC5CDR Disease, and BC5CDR Chemical datasets, we compare our proposed model's score with previous researches. Bioinformatics datasets are reported by top one recommendation accuracy. Given the biomedical entity in the train set, entities are matched with the most similar entities in datasets. If the query entity and target entity share the same concept ID, it is considered correct. The financial NEN dataset is a pairwise NEN matching corpus. For evaluations on the financial NEN dataset, models that are used in evaluations distinguish whether two named entity pairs share identical meanings or not. We also perform the qualitative analysis to assess models' weaknesses.

4.4.1 Quantitative Analysis: Bioinformatics

Table 4.5: Bioinformatics named entity normalization performance test

	NCBI Disease	BC5CDR Disease	BC5CDR Chemical
Sieve-Based [17]	84.7	84.1	90.7
Taggerone [48]	87.7	88.9	94.1
CNN Ranking [52]	86.1	-	-
NormCo [88]	87.8	88.0	-
BNE [66]	87.7	90.6	95.8
BERT Ranking [32]	89.1	-	-
TripletNet [59]	90.0	-	-
BioSyn [83]	90.7	92.9	96.6
BioSyn with TF-IDF [83]	91.1	93.2	96.6
Proposed Model	91.7	93.4	96.7
Proposed Model with TF-IDF	**92.1**	**93.7**	**96.7**

Table 4.5 shows a performance comparison between our proposed model and previous state-of-the-art models. The best scores are boldfaced and the second best scores are underlined. We also train and report our model's performance with TF-IDF vectors added to the vanilla embedding vectors that is illustrated on the previous state-of-the-art model, BioSyn [83]. For three bioinformatics datasets, our proposed model achieved the highest accuracy. Our model showed the highest performance increase by 1.0% in the NCBI Disease corpus. For BC5CDR Disease and BC5CDR Chemical corpus, the performance increase compared the previous state-of-the-art model is 0.5% and 0.1%, respectively.

The NCBI Disease corpus is a comparatively harder dataset based on the performance of other models. We conclude that there there is a significant to increase the accuracy in a relatively lower performing dataset. The previous model already performs excellently on the the BC5CDR corpus with accuracy scores of 93.2% to 96.6%. Especially, the earlier NEN models on BC5CDR Chemical dataset already achieved over 90% and the updates on the performance is decreased yearly, the improvement on this specific dataset might reached the plateau.

4.4.2 Quantitative Analysis: Finance

Table 4.6: Precision, Recall, F-Score, Accuracy of models

	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)
Edit Distance	43.12	63.35	51.31	62.60
BERT	62.92	82.17	71.27	76.81
Siamese GCN	79.00	82.16	80.55	82.56
Siamese BiLSTM	75.96	89.98	82.38	85.15
BioSyn [83] by top1	99.73	77.77	87.39	88.75
BioSyn [83] by top3	99.38	89.79	94.34	94.60
Proposed Model by top1	**99.82**	90.88	95.14	95.35
Proposed Model by top3	99.68	**98.03**	**98.85**	**98.85**

Table 4.6 shows the performance of each model we test. The evaluation metrics are expressed as follows

$$\begin{aligned} precision &= \frac{tp}{tp+fp} \\ recall &= \frac{tp}{tp+fn} \\ F1 &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \end{aligned} \tag{4.4}$$

False positive indicates that two entities should not be matched, but our proposed model decided to link two entities. False negative indicates that two entities should be matched, but our proposed model failed to link two entities.

For practical use in the NEN model in the finance domain, a model with higher precision should be rewarded more. In practice, a model with higher precision will reduce the burden for practitioners' tasks by giving more reliable entity-matching results. A model with higher precision will reduce time double-checking the validity entity pairs marked as matched.

Edit Distance had the lowest score along with all performance evaluation indicators. Graph Convolutional Network we use for the experiments adopts the BERT vector as entity node features. BERT and GCN have a similar recall, but GCN has higher precision, which brings higher F-score and accuracy compared with BERT. Our proposed model achieved the highest precision, F-score, and accuracy. Among all the models, our proposed model is the only model with a precision score over 90%. Therefore, our proposed model is the most suitable for practical use.

4.4.3 Qualitative Analysis

Error Analysis

In error analysis, entities for which accurate recommendations are not made are reported. Through error analysis, we aim to recognize the pattern of cases where recommendations are not properly made.

Table 4.7: Error analysis on three biomedical NEN datasets

	Query Entity	Retrieved Synonym Entity
NCBI Disease	encephalopathy	aids encephalopathy
	nail dystrophy	twenty nail dystrophy
	cdm	cdmd
	copper overload	copper deficient
	g m2 gangliosidosis	g m2 gangliosidosis type ii
BC5CDR Disease	lung mass	liver mass
	hypoactivity	hyperactivity
	htn	htx
	thrombocytopenia type ii	thrombocytopenia 2
	chronic liver disease	chronic hepatitis
BC5CDR Chemical	inorganic as	chemicals inorganic
	alcohol nicotine	alcohol nicotinyl
	dph	ddph
	naoh	natrolite
	myo inositol 1 phosphate	myo inositol 1 3 6 triphosphate

Table 4.7 lists the errors in three bioinformatics NEN datasets. Our proposed model achieves approximately 90% accuracy for all three datasets. However, finding the synonyms for short abbreviations such as “cdm”, “htn”, and “dph” seems relatively harder. In addition, if there exist longer overlapping strings, the performance of the model is degraded.

Table 4.8: Error analysis on financial NEN dataset

False Positive	
Entity 1	Entity 2
Park Hyatt	(Marriott, Hyatt, Hilton and AccorHotels)
AT & T Acquisition	AT & T Corp. ' s (ATTC)
Bureau of Indian Affairs (BIA)	Balanced Budget Act of 1997 (BBA)
Garmin Corporation	Garmin GTN Xi
Windows Server	Microsoft Teams
False Negative	
Entity 1	Entity 2
(LTE)	4G
EU Member State	European Union Member States
RPS	Renewable Portfolio Standards (RPS)
737-800 Boeing 737	(B737)
Cyber Security Regulation	Privacy and Cyber Security Regulation

Financial NEN datasets are constructed using entity pairs. Our model predicts whether two entity pairs are matched or not. Table 4.8 is divided into false positive lists and false-negative lists. By examine the false-positive lists, entities with similar meanings or with matching strings are often predicted positive while the actual label is negative.

We also examine the false negatives. Matching named entities with parenthesis and abbreviations is the part where our model's prediction is relatively unstable. Entity pairs such as “LTE” and “4G” can be one of the most difficult to predict as positive because the intrinsic meaning of “LTE” and “4G” requires common sense. Even our model is based on BERT, which captures the semantic meaning from the sentences where named entities are excerpted, using the common sense beyond the information presented in surrounding sentences can be limited.

Named Entity Normalization Result According to Training Progresses

Table 4.9: Named entity normalization result for epoch 0, epoch 1, and epoch with highest accuracy

	NCBI Disease: ***c2 deficiency***		
	Epoch 0	Epoch 1	Epoch with Highest Accuracy
Top 1	**c2 deficiency**	**c2 deficiency**	**c2 deficiency**
Top 2	c3 deficiency	c6 deficiency	**c2 deficient**
Top 3	t2 deficiency	c3 deficiency	**hereditary c2 deficiency**
Top 4	c5 deficiency	**c2 deficient**	**type ii c2 deficiency**
Top 5	cpox deficiency	c4 deficiency	**type i c2 deficiency**
	BC5CDR Disease: ***failing left ventricle***		
	Epoch 0	Epoch 1	Epoch with Highest Accuracy
Top 1	tumor cerebral ventricle	dysfunction left ventricular	**left sided heart failure**
Top 2	cerebral ventricle tumor	**left sided heart failure**	**heart failure**
Top 3	tumors cerebral ventricle	remodeling left ventricular	**cardiac failure**
Top 4	syndrome slit ventricle	hypertrophy left ventricular	**heart failure left sided**
Top 5	ventricle tumor cerebral	outflow obstruction left ventricular	**right sided heart failure**
	BC5CDR Chemical: ***vincristine sulfate***		
	Epoch 0	Epoch 1	Epoch with Highest Accuracy
Top 1	**vincristine sulfate**	**vincristine sulfate**	**vincristine sulfate**
Top 2	**sulfate vincristine**	**vincristine**	**vincristine**
Top 3	vinblastine sulfate	voacristine	**sulfate vincristine**
Top 4	sulfate vinblastine	**leurocristine**	**vincristin**
Top 5	riboflavin 3 sulfate	ergocristine	**vincristin medac**
	Financial NEN: ***Polo Ralph Lauren Children***		
	Epoch 0	Epoch 1	Epoch with Highest Accuracy
Top 1	Pinky Swear Foundation	**Polo Golf Ralph Lauren**	**Polo Ralph Lauren**
Top 2	Bath & Body Works Canada	**Polo Ralph Lauren**	**Polo Ralph Lauren Children, Chaps**
Top 3	Ticketmaster North America	Siemens Medical Solutions USA	**Polo Golf Ralph Lauren**
Top 4	LIP-BU TAN	Mojo Networks, Inc.	Lilly International
Top 5	Coca-Cola Life	**Polo Ralph Lauren Children, Chaps**	**Polo / Lauren Company, LP**

As the training epochs increase, recommendations become more accurate. We randomly selected entities from four datasets we tested. Top 5 recommendations for the selected entities are provided for epoch 0, epoch 1, and epoch with best result in Section 4.4.1 and Section 4.4.2.

Table 4.9 shows how recommendations change as training progress. Entities after each dataset are the examples excerpted (c2 deficiency, failing left ventricle, vincristine sulfate, and Polo Ralph Lauren Children), and bold-underlined entities are the entities with the same concept ID as the query entity. Throughout the datasets, at epoch 0, the recommended entities differ greatly from the concept ID of the query entity. As the model is trained, the recommendation becomes more accurate in epoch 1. At the epochs in which the highest accuracy for the datasets is achieved, true synonyms for query entities are successfully selected.

Based on our experiments, our proposed model has the highest precision, recall, F1 score, and accuracy. Qualitative analysis shows that our proposed model also gives the most stable results by achieving over 98% on the four evaluation metrics.

4.5 Chapter summary

We introduce Edge Weight Updating Neural Network for NEN. NEN to match extracted named entities with homogeneous identity is pivotal for many text mining tasks. We tested our model on three widely used NEN datasets, NCBI Disease, BC5CDR Disease, and BC5CDR Chemical. We also generated the NEN dataset for the finance domain. Next, we verify our model's performance for general NEN applications.

The main contribution of this study are as follows. Our proposed model successfully links named entities with the same meanings with different surface forms. The proposed model performs best among previous NEN models. We test our model not only for bioinformatics datasets in which NEN researches are more active but also for financial NEN datasets. According to the performance of the NEN corpus in two distinct fields, our proposed model proves the efficacy for general NEN applications.

Similar to many other NEN models, the performance of linking named entities with abbreviations is comparatively lower. Matching abbreviations more accurately is one of the future works. The neural network model with our proposed Edge Weight

Updating objective function performs better than other models. Providing the more general guideline for the number of training epochs and increasing the training stability is one of the future research topics.

Chapter 5

Building knowledge graph using named entity recognition and normalization models

5.1 Background

Recent advances in technology have been actively witnessed through a wide range of venues, one of which includes the patent claims. Patent claims contains information on the new breakthroughs at the forefront of the industry and academia in the rawest form, which may potentially help solve various tasks such as discovering contemporary technological trends, forecasting future developments in specific domains, evaluating ideas for R&D investment decisions, identifying competitors in the technological horse-races, or developing strategic technological planning [1].

The rapid speed and the vast volume of patent filings have been worsening the challenge of distillation of useful information from the claims, which is calling for the automation, at least in part, of patent analysis. Until recently, research on patent analysis has generally involved extracting technology trees based on the bibliographic connections of the claim filings [91] or extracting keywords using text mining techniques [40, 19, 20]. While these keyword-based approaches have provided meaningful insights on the current technological developments, only few attempts have been made to extract a more complicated form of information from the patent filings, such as named entities. Named entities, which include technological concepts, specific techniques used, names of the devices or the end products, and the associated company names, are of a significant importance for richer and deeper understanding of the innovations and technology underlying the patent filings.

Moreover, past efforts have failed to provide information on the intricate connectivity among the concepts extracted from the patent-related documents. For example, a well-designed keyword detection models may successfully determine the term “Gate-All-Around” to be the arising keyword alongside the word “transistors” from the patent filings within the field of semiconductor devices, yet it will not be able to show through which patent documents and other keywords these two phrases are interconnected. Furthermore, the conventional NLP approach will parse the terms “Gate-All-Around” and “GAA” separately as two independent terms, leaving the task of recognizing them as the same entities to additional human efforts.

In this study, we address the issues of interconnecting key technological concepts and matching the same entities appearing in different forms by constructing a semiconductor-related patent knowledge graph from patent filings using the NER and NEN models with a novel edge weight updating neural network. More specifically, we constructed the NEN dataset based on the patent documents. We fine-tuned the NER model [14] using Huggingface’s Python repository, pre-trained with the CoNLL-2003 NER dataset [74]. Our BERT token concatenator for NER tasks provides more complete named entity phrases. We propose a state-of-the-art NEN model with an edge weight updating neural network with triplet loss to extract named entities and connect them through the semiconductor related patent documents and present them in the form of a knowledge graph. Our proposed NEN model achieves the highest performance for not only the conventional candidate retrieval task in NEN but also pairwise named entity matching task. Extensive experiment results show that our proposed approach performs, against the conventional keyword extraction models frequently employed in patent analysis, very competitively, especially for the NEN and document retrieval tasks. We also show that our knowledge graph construction method is robust to the out-of-vocabulary problem. Finally, we further contribute to the existing literature by releasing our semiconductor-related patent knowledge graph online, available for all non-commercial purposes.

5.2 Proposed model

Our approach consists of three major components: (1) NER using the Huggingface's pre-trained NER model; (2) NEN by relying on our novel edge updating neural network; and (3) construction of the semiconductor-related patent knowledge graph. Figure 5.1 shows the overall framework of the proposed method.

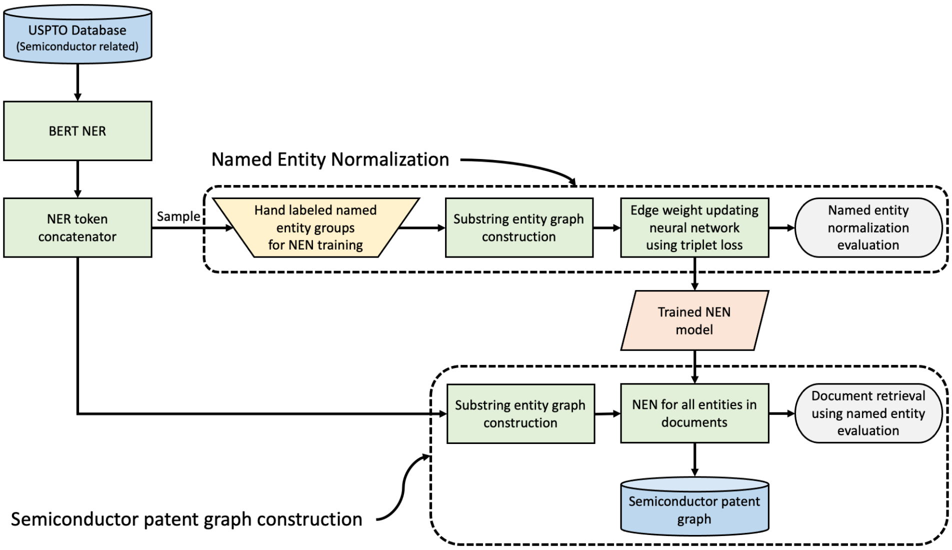

USPTO Database
(Semiconductor related)
BERT NER
Named Entity Normalization
NER token concatenator
Sample
Hand labeled named entity groups for NEN training
Substring entity graph construction
Edge weight updating neural network using triplet loss
Named entity normalization evaluation
Trained NEN model
Substring entity graph construction
NEN for all entities in documents
Document retrieval using named entity evaluation
Semiconductor patent graph
Semiconductor patent graph construction

Figure 5.1: Overall framework

5.2.1 Named entity normalization

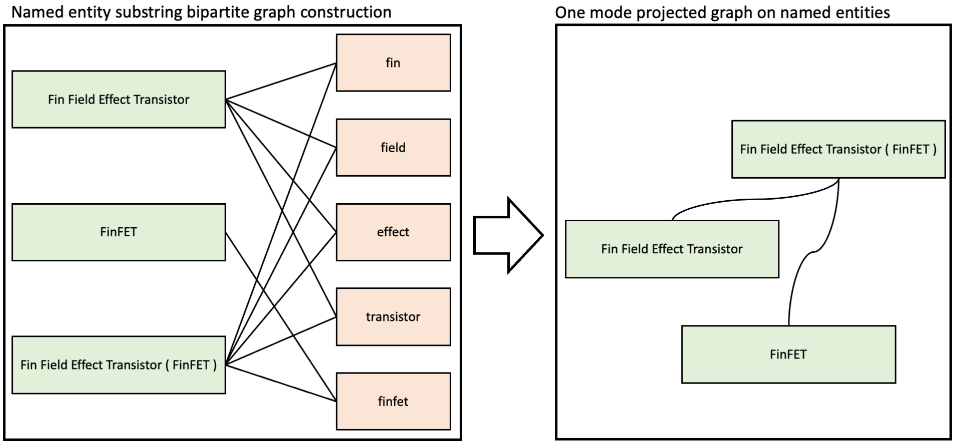

Named entity substring bipartite graph construction
Fin Field Effect Transistor
FinFET
Fin Field Effect Transistor (FinFET)
fin
field
effect
transistor
finfet
One mode projected graph on named entities
Fin Field Effect Transistor (FinFET)
Fin Field Effect Transistor
FinFET

Figure 5.2: Named entity graph construction based on named entities' substrings

With the named entities concatenated as described in Section 3.2.2, we initialize the construction of the named entity graph by connecting the substrings of the extracted named entities, and the process is summarized graphically in Figure 5.2.

Edge weight normalization

Our named entity graph construction proceeds as follows: first, we parse the named entities, resulting from the concatenation stage, using the whitespaces. For example, the entity "Fin Field Effect Transistor", after the parsing, will result in the token pieces "fin", "field", "effect", and "transistor". Please note that we exclude the punctuation and the common stopwords during the parsing process. We repeat the parsing process on every named entity and construct a bipartite graph with the named entities as one group, and the associated substrings resulting after the parsing, as another, as illustrated in the left panel of Figure 5.2. Then, we one-mode project the bipartite graph on the named entity level. The resulting network consists of named entities as its nodes and the number of shared substrings between each pair of named entities as the edges, as depicted in the right box of Figure 5.2.

At this time, the entities “FinFET” and “Fin Field Effect Transistor” are not directly connected by an edge, but only indirectly via the common neighbor node “Fin Field Effect Transistor (FinFET)”. Given that these indirectly connected components of entities imply similar ideas, our goal is to determine and connect nodes of the exact same concept/definition. To do so, we compute the relative strength between the two named entities by normalizing the edge attributes based on Equation 5.1 as follows:

$$\begin{aligned} A &= (W_s, W_t, W_m) \\ W_s &= \frac{w_{s,t}}{max(\{w_{s,i} \mid i \in C_s\})} \\ W_t &= \frac{w_{s,t}}{max(\{w_{t,j} \mid j \in C_t\})} \\ W_m &= W_s \cdot W_t \end{aligned} \tag{5.1}$$

where $A = (W_s, W_t, W_m)$ denotes the edge attributes and W_s, W_t the normalized strength between the two named entities given the source entity and the target entity, respectively. To dilute the over-fitting calculations for the relatively shorter named entity and the undermining calculations for the relatively longer named entity, W_m is set to the product of W_s, W_t. The number of shared substrings between the named entity i and j is denoted by $w_{i,j}$, and the connected components of the named entity are i, C_i.

Edge weight updating neural network using triplet loss

We first learn the named entities using the pre-trained BERT embedding model, and then we fine-tune the parameters with our novel edge weight updating neural network [30] using triplet loss [76]. The triplet loss function we employ is mathematically defined by Equation 5.2, where a, p, and n are the anchor and the positive and negative vectors, respectively:

$$\begin{aligned} \mathcal{L}(a,p,n) &= max\{0, d(a,p) - d(a,n) + margin\} \\ where&, \\ d(x,y) &= \|x-y\|_2 \end{aligned} \tag{5.2}$$

For training the model using the triplet loss function, several issues need to be considered. First, because the loss function takes triplets as its input, the number of triplet combinations explodes as the size of the data increases. At the same time, the model performance is found to be sensitive to the quality of the triplets used during the training. In other words, a selection of adequate triplets for the training is necessary.

Previous studies have suggested promising solutions to this challenge. Hermans et al. [28] proposed the batch-hard triplet loss, which chooses the most definitively positive and negative samples when constructing the triplets for the online training. Yu et al. [92], averaged the negative and positive samples instead of constructing sample-to-sample triplets.

In this study, we adopt the batch-hard triplet loss approach as implemented in [28]. Furthermore, we make use of a scarcely labeled dictionary of named entities with its variant identities as supplementary data source because the use of external information during the training process exploiting the triplet loss function leads to a significant increase in the quality of the positive and the negative samples [30].

We begin our training by using the graph resulting from Section 5.2.1. The similarity between the two BERT vectors is then determined by computing the inner products. After the first epoch of training, the positive and negative samples are determined as follows: among the entities connected on the network, the entity pair with the greatest similarities, yet labeled as unmatched (that is, labeled as "0") in the dictionary, is considered as the negative input of the triplet loss. In contrast, the entity pair labeled as matched (labeled as "1"), yet the inner product of its BERT vectors, is the lowest and is considered as the positive input of the triplet loss. The positive and negative samples are then consumed as inputs in the next training epoch, given the BERT vector similarity of the previous epoch among connected entities in the substring graph.

Mathematically speaking, let the set of named entities be denoted by $N = [entity1, entity2, \dots]$; and the connected entities of the anchor entity a with the positive and negative labels, $C_{a_{pos}}$ and $C_{a_{neg}}$, respectively. Then, the total training loss can be expressed as Equation 5.3.

$$
\begin{aligned}
\mathcal{L} = \sum_{a \in N} max\Big[&0, min\{d(f(a), f(x_p)) \ \mid x_p \in C_{a_{pos}}\} \\
&- max\{d(f(a), f(x_n)) \mid x_n \in C_{a_{neg}}\} + margin\Big] \\
where, & \\
d(x, y) &= \|x - y\|_2 \\
f(x) &= BERT(x)[CLS]
\end{aligned}
\tag{5.3}
$$

We further illustrate our approach using an example. Consider an anchor entity, “Fin Field Effect Transistor (FinFET)”, and another one, “FinFET”, from the same entity group but that shares only one substring, whereas “Metal Oxide Semiconductor Field Effect Transistor (MOSFET)”, which should be placed in a different entity group, still shares the substrings, “Field”, “Effect”, and “Transitor”. In this case, “FinFET” will serve as the positive input for the entity “Fin Field Effect Transistor (FinFET)”, while “Metal Oxide Semiconductor Field Effect Transistor (MOSFET)” will take the place of the negative sample.

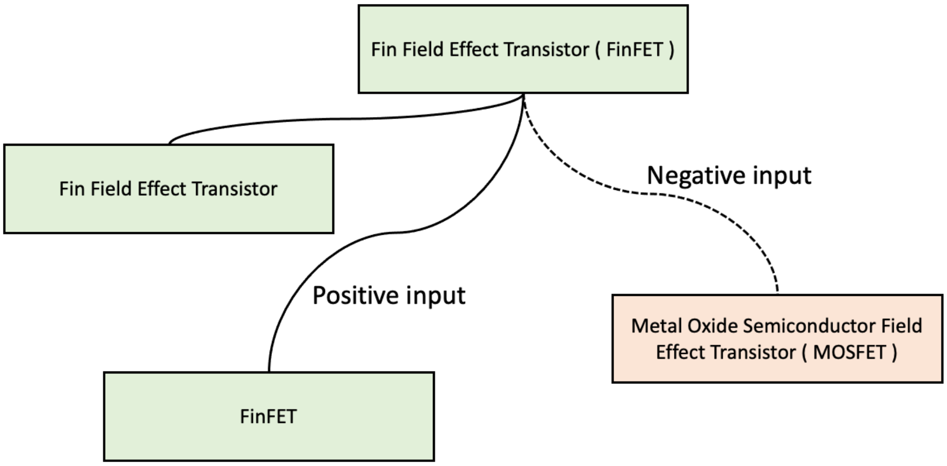

Fin Field Effect Transistor (FinFET)
Fin Field Effect Transistor
Negative input
Positive input
Metal Oxide Semiconductor Field Effect Transistor (MOSFET)
FinFET

Figure 5.3: Edge weight updating neural network with triplet loss

Figure 5.3 shows the example of the positive and negative inputs to train the edge weight updating neural network with triplet loss.

5.2.2 Construction of the semiconductor-related patent knowledge graph

The semiconductor-related patent knowledge graph is completed using the following process. We train a simple regressor to determine whether the two given named entity pairs of the fine-tuned BERT embeddings, first appearing in the initial graph resulting from Section 5.2.1, have survived in the trained model in Section 5.2.1. All of the linked entities in the substring named entity graph are tested and updated. Then, the connected components of the final graph are considered the unique named entity groups. These groups are expressed as separate nodes of a different mode, which corresponds to the named entity groups. Finally, the semiconductor patent knowledge is completed by linking the patent document, in which the named entity appeared.

5.3 Experiment results

5.3.1 Comparison models

We tested our proposed approach against conventional and standard text mining models: Word2vec [58], Glove [65], Fasttext [34], and BERT [14]. SciBERT [6] is the variant of the original BERT model, pre-trained with scientific text, which might be more suitable for patent-related analysis. Hence, we also included SciBERT in our experiment. BioSyn [83] is one of the state-of-the-art NEN models. The Biomedical documents were used for training in the original BioSyn paper. We trained the BioSyn model with our patent NEN dataset and compared the performance with other models including our proposed model. The weighted averaged vectors of each word embedding model were used for the embedding of the named entities. Table 5.1 summarizes the basic characteristics of the baseline models in terms of the NEN and document retrieval tasks.

Table 5.1: Models used for the evaluation

Models	Descriptions
Word2vec [58]	Word2vec is one of the most widely used NLP models. In our research, pre-trained Word2vec vectors were used. More specifically, word2vec-google-news-300[1] was used, which consists of 3 million words using Google News. The dimension of each embedding vectors is 300.
Glove [65]	Word2vec is constructed to predict the neighboring words given a window size. However, in Glove the dot product of the embedding of the target word and that of neighboring word matches the co-occurrence of words in the corpus. We used the pre-trained Glove vectors, glove-wiki-gigaword-300[2]. The pre-trained model consists of 400,000 word vectors trained with Wikipedia 2014 data and Gigaword 5[2]. The dimension of each word embedding vectors is 300.
Fasttext [34]	Fasttext utilizes a training mechanism similar to that of Word2vec. Unlike Word2vec and Glove, Fasttext splits the words into subwords tokens. Fasttext is known to be the more robust NLP model when handling out-of-vocabulary problems. In technical documents, where new terminologies frequently appear, Fasttext can be the more suitable model. The pre-trained Fasttext model, fasttext-wiki-news-subwords-300[3], was used in our research, which contains a million word vectors. The dimension of each word embedding vectors is 300. The pre-trained model was trained with the Wikipedia 2017, UMBC webbase corpus, and statmt.org news datasets[3].
BERT [14]	BERT is one of the state-of-the-art models for various NLP tasks. However, for our specific tasks, the BERT model has a limitation in capturing the morphological similarity between entity pairs. We used pre-trained BERT vectors[4] with size of 768 and train a simple MLP classifier with batch size of 4096 to determine the linkage between entity pairs.
SciBERT [6]	SciBERT uses the BERT model architecture. The model has been fine-tuned with various scientific documents. For specific NLP tasks, such fine-tuned models shows higher performance compared to the vanilla BERT model. As the structure of SciBERT is the same as that of the BERT model, the embedding dimension is same as that of the BERT model.
BioSyn [83]	BioSyn is one of the state-of-the-art NEN models. In the original report, the NEN was concentrated on bio-medical documents and used BioBERT[49] for the pre-trained embedding model. BioSyn implements marginal maximum likelihood (MML) for the objective function. We trained the BioSyn model with our patent NEN dataset. As the aim of the BioSyn model is NEN, evaluations of BioSyn in information retrieval tasks in Section 5.4.1 and Section 5.4.1 are excluded.

[1] https://code.google.com/archive/p/word2vec/

[2] https://nlp.stanford.edu/projects/glove/

[3] https://fasttext.cc/docs/en/english-vectors.html

[4] https://github.com/google-research/bert

5.3.2 Parameter settings

The experiments were executed using an Intel Core-i9-10940X CPU with 128GB of memory and three NVIDIA GeForce Titan RTX GPUs. For training the edge weight updating neural network using triplet loss as described in Section 5.2.1, the batch size was set at 64, and the learning rate was 10^{-5} using the Adam optimizer [41] with weighted decay [57]. The model was trained for 50 epochs. We report the best performing model out of all the results obtained after each epoch.

5.4 Results

5.4.1 Quantitative evaluations

Named entity normalization: candidate retrieval

Many NEN models from previous studies are evaluated by the candidate retrieval tasks [52, 66, 32, 83]. We evaluated the performance of the candidate retrieval for NEN with various models. An evaluation was conducted to validate the efficacy under the same conditions as those for the previous NEN models including BioSyn [83], the current state-of-the-art NEN model. The evaluation was reported based on whether the group id of query entity and the group id of the most similar entity from the dictionary dataset were the same. The performance of the models is presented in Table 5.2.

Table 5.2: Named entity normalization by candidate retrieval performances

	Accuracy
Word2vec[1]	89.71%
Glove[2]	73.88%
Fasttext[3]	85.81%
BERT [4]	64.17%
SciBERT [4]	57.33%
BioSyn	92.52%
Our Model	**97.46%**

[1] Out of vocabulary: 1,074.

[2] Out of vocabulary: 1,802.

[3] Out of vocabulary: 937.

[4] Query vectors are smoothed by the entity group. The smoothing is conducted by averaging all entity vectors in the group.

BERT [14] and SciBERT [6] models not specifically trained for NEN tasks. We utilized the similarity ranking model described in the study of Ji et al. [32], but the retrieval of a single entity was unsuccessful for many entities. Smoothing the dictionary vectors by averaging the entity vectors in the named entity group gave relatively higher accuracy. Among the models we tested, our proposed model achieved the highest performance in candidate retrieval tasks for NEN.

Named entity normalization: pairwise matching

the model performances were tested by precision, recall, f-score, and accuracy, computed as defined in Equation 5.4, which are standard metrics for evaluating the pairwise named entity matching tasks.

$$\begin{aligned} precision &= \frac{tp}{tp + fp} \\ recall &= \frac{tp}{tp + fn} \\ F1 &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \end{aligned} \tag{5.4}$$

We detected connected components from the semiconductor-related patent knowledge graph as the unique named entity groups. Thus, we evaluated to which extent these connected components matched well as compared to the ground truth groups by computing the V-measure [73]. The V-measure calculates the harmonic mean of the other two widely used clustering evaluation metrics, homogeneity and completeness, to assess the range of the overlap between the given clusters and the ground truth grouping. The mathematical definition is expressed by Equation 5.5.

$$V = \frac{(1 + \beta) \cdot homogeneity \cdot completeness}{\beta \cdot homogeneity + completeness} \tag{5.5}$$

In our evaluation, we assumed that the weight was equal across homogeneity and completeness by setting $\beta = 1$.

Table 5.3 reports the performance of our proposed approach against the baseline models. The results show that our model beats the conventional embedding methods in almost every case. In particular, only our model achieved over 90% in precision and recall in the pairwise entity matching tasks. By scoring over 0.97 in V-measure, the named entity groups constructed by our proposed model highly resembled the ground truth named entity groups. SciBERT with the substring graph showed the best performance in terms of recall, yet compared to our model, the measure is very close, and it differs by 0.1%. Such outstanding performances against the baseline models, we believe, is largely owed to the out of vocabulary problems. To support our claims, we additionally report the number of out of vocabularies at the end of Table 5.3. Word2vec, Glove, and Fasttext are known to perform relatively less robust to the words unseen during the training process, hence deteriorating performance when they met newly rising concepts. Given the recent fast-paced technological developments, however, handling out-of-vocabulary concepts is critical in scientific documents. The experiment results show that our proposed model performs well in such cases and works robustly when faced with newly introduced words never seen before.

Table 5.3: V-measure, precision, recall, F-score, and accuracy of models

	V-measure	Precision	Recall	F-score	Accuracy
Word2vec[1]	0.5810	65.53%	65.01%	65.27%	66.11%
Word2vec[1] with substring graph	0.7579	81.79%	86.95%	84.29%	84.12%
Glove[2]	0.5101	61.45%	64.15%	62.77%	65.62%
Glove[2] with substring graph	0.7528	81.43%	86.71%	83.99%	85.06%
Fasttext[3]	0.6013	69.85%	62.55%	66.00%	68.76%
Fasttext[3] with substring graph	0.8298	82.91%	90.58%	86.58%	86.38%
BERT	0.4922	68.66%	76.11%	72.20%	70.60%
BERT with substring graph	0.5943	82.04%	**92.32%**	86.88%	86.01%
SciBERT	0.5091	72.59%	72.14%	72.37%	72.37%
SciBERT with substring graph	0.7644	86.51%	88.79%	87.63%	87.44%
BioSyn	0.5824	73.24%	74.75%	73.99%	73.64%
BioSyn with substring graph	0.6688	80.96%	89.48%	85.01%	84.17%
Our Model	**0.9787**	**94.45%**	91.92%	93.17%	93.24%
Our Model with substring graph	**0.9787**	**94.45%**	92.20%	**93.31%**	**93.37%**

[1] Out of vocabulary: 1,074.
[2] Out of vocabulary: 1,802.
[3] Out of vocabulary: 937.

Document retrieval from the named entities

In this section, we report the performance of our proposed model in relation to the document retrieval task. To be as fair as possible, we restrained from querying named entities as we conducted the test. For the competing embedding models such as Word2vec, Glove, Fasttext, BERT, and SciBERT, the representations of each entity and each document was computed as the weighted average of all the tokens associated with the respective named entity or the documents. BioSyn is a model that specifically focuses on NEN tasks, so BioSyn is not used for the retrieval tasks. As for our proposed model, because our end-product has the form of a network, we take advantage of the structural characteristics of the knowledge graph. When given the query, we return the document with the highest edge weight connected to the given named entity's group. We test the relevance of the document recommendations in response to the given query based on whether the named entity and the retrieved documents are from same CPC group (total: 50) and CPC subgroups (total: 449). The performances of each model are reported in Table 5.4.

Table 5.4: Accuracy of document retrieval from the named entities

	Accuracy for CPC group(50)	Accuracy for CPC subgroups(449)
Word2vec	70.70%	51.58%
Glove	68.14%	50.48%
Fasttext	48.31%	33.79%
BERT	67.26%	53.83%
SciBERT	62.09%	46.78%
Our Model	**85.78%**	**77.46%**

Across CPC groups and subgroups, our proposed model reports the highest accuracy. Our model achieved over 77% accuracy on retrieving the relevant documents with respect to the CPC subgroups. This, in particular, is an impressive result given that there were 449 subgroups. Due to the granularity of the sub-groupings, all of the other baseline models suffer gravely in terms of accuracy.

Named entity retrieval from the documents

Retrieving the relevant named entities from patent documents is another important task. The most related named entity was retrieved using the following procedures. For Word2vec, Glove, Fasttext, BERT, and SciBERT, the embedding vectors of the named entities that appeared in each document were averaged. Based on the embedding obtained for each document, the named entity with the highest similarity was recommended. As for our proposed model, we returned the named entity within the group that had the highest connected edge weight to the given document's named entities. For the named entity recommendations, the named entities that appeared directly in the document were excluded from the candidates. We evaluated the performance of the named entity recommendations in response to the given query based on whether the documents and the retrieved named entities were from same CPC group (total: 50) and CPC subgroups (total: 449). The performances of each model are reported in Table 5.5.

Table 5.5: Accuracy of the named entity retrieval from the patent documents

	Accuracy for CPC group(50)	Accuracy for CPC subgroups(449)
Word2vec	84.96%	72.78%
Glove	83.65%	68.53%
Fasttext	81.44%	67.73%
BERT	85.86%	64.28%
SciBERT	77.04%	61.05%
Our Model	**91.65%**	**83.68%**

Our proposed model reports the highest accuracy for the named entity retrieval tasks. Our model achieved over 83% accuracy on retrieving the relevent named entity with respect to the CPC subgroups. The second best performing model was word2vec, which showed an accuracy of 73%, and the difference in performance compared to that of our model was approximately 10%. The proposed model has a significant improvement in performance compared to that of other models, and it can provide insights by accurately retrieving the related named entities from the document.

5.4.2 Qualitative evaluations

Error analysis on the pairwise named entity normalization

As the quality of the semiconductor-related patent knowledge graph relies heavily on the performance of the NEN process, we report the result of the error analysis we conducted on the pairwise NEN in this section. More particularly, we report the false positive examples in Table 5.6 and the false negative examples in Table 5.7 on the pairwise NEN tasks with the model's confidence.

Table 5.6: Examples of false positives of pairwise named entity normalization task

Entity 1	Entity 2	Confidence[1]
Silicon germanium (SiGe) ,	Silicon Nitride (SiN) .	0.5400
(Organic Light Emitting Diode) or	Light - Emitting - Diode (LED) is	0.6526
(PVD) processes	(CVD) processes	0.6303
Digital Subscriber Line (DSL) ,	Digital Signal Processor (DSP) ,	0.6072
DC - AC inverter	(AC) power	0.5275

[1] Confidence of our model to predict the entity pair as the label 1(the matching entity pair).

Table 5.7: Examples of false negative of pairwise named entity normalization task

Entity 1	Entity 2	Confidence[1]
Cobalt (Co) based	Cobalt (Co) material	0.5504
Teflon (PTFE) ,	Teflon®	0.6565
FPC (Flexible Printed Circuit) ,	(Flexible Printed Circuit) .	0.5798
Linux OS	Linux™	0.5720
APPLE iPad .	(e . g . , iPad®) ,	0.6894

[1] Confidence of our model to predict the entity pair as the label 0(the non-matching entity pair).

In general, both false positive and false negative results have relatively low confidence. This implies that, when constructing the semiconductor patent named entity graph, connecting the undesired entity pairs can be prevented by connecting the entities with higher confidence.

5.4.3 Knowledge graph visualization and exemplary investigation

By training the NEN model as discussed in Section 5.2, with our hand-labeled dataset as described in Section 3.2.2, we have successfully recognized 69,812 named entities and connected the entity pairs with a confidence over 0.999 to maximize precision. After pruning the false positive links, we ended up with a knowledge graph with 25,938 named entities assigned to the total of 8,525 unique named entity groups. The overall statistics of the semiconductor patent named entity knowledge graph are listed in Table 5.8.

Table 5.8: Statistics of the semiconductor patent named entity knowledge graph

Types	Number of nodes	Number of edges
Patent document nodes	34,356	—
Named entity nodes	25,938	—
Named entity group nodes	8,525	—
Total nodes	68,819	—
Total edges	—	297,542

We present the graphical visualization of the entire knowledge graph in Figure 5.4. The resulting graph may also be accessed freely online via an interactive environment, available for all non-commercial purposes [1].

The purple nodes represent the patent documents; the green nodes, the named entity groups, and; the orange nodes, the associated named entities.

[1] https://sjeon7.github.io/Semiconductor_Patent_Named_Entity_Graph/network/

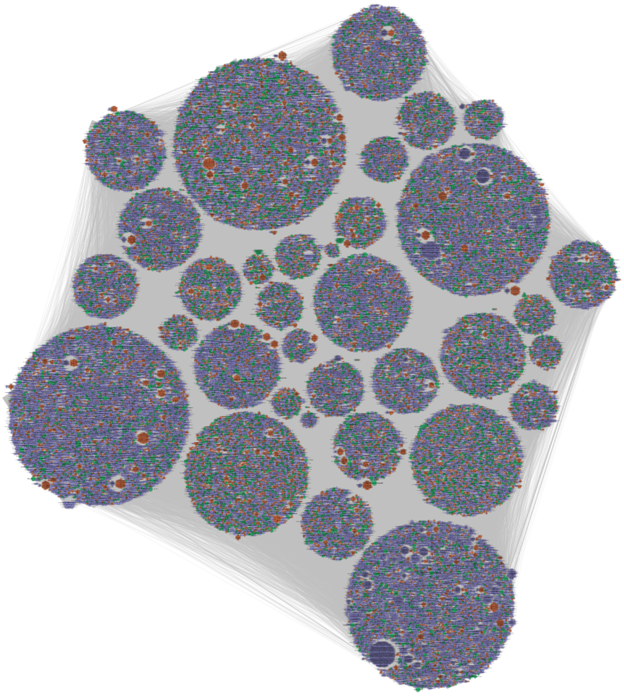


Figure 5.4: Semiconductor patent named entity knowledge graph

As it is difficult to distinguish visually the graph with almost 70,000 nodes, we selected three named entity groups, “USB type C”, “deep neural network”, and “Samsung Galaxy S” and report the resulting subgraphs in Figure 5.5, Figure 5.6, and Figure 5.7, respectively, for demonstrative purposes. As can be easily observed in these subgraphs, the named entity nodes of similar technological concepts are successfully grouped. For example, the terms “Universal Serial Bus” is well connected to the entities “USB” and “USB Type C” in the subgraph in Figure 5.5, and the patent connected to those named entity nodes well encompasses these terms.

A similar pattern is observed for the subgraph reported in Figure 5.6, which shows the connection between the original phrase, “deep neural network”, with its abbreviation, “DNN”, correctly established.

The example reported in Figure 5.7 shows that our model, in addition to the technological jargon, also successfully extracted and connected brand and product names.

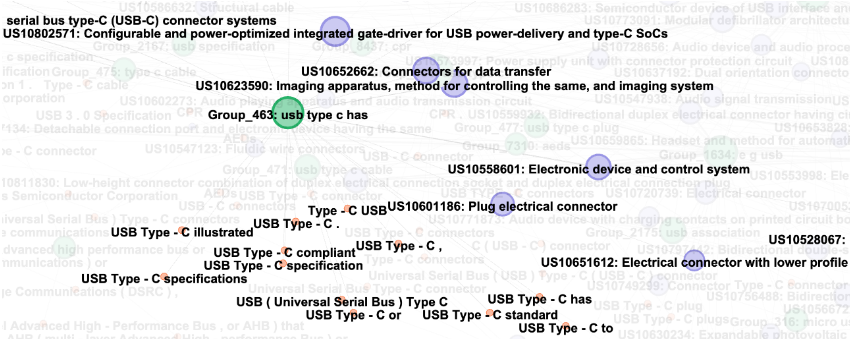

serial bus type-C (USB-C) connector systems
US10802571: Configurable and power-optimized integrated gate-driver for USB power-delivery and type-C SoCs
US10652662: Connectors for data transfer
US10623590: Imaging apparatus, method for controlling the same, and imaging system
Group_463: usb type c has
US10558601: Electronic device and control system
Type - C USB
US10601186: Plug electrical connector
USB Type - C .
USB Type - C illustrated
USB Type - C ,
US10528067:
USB Type - C compliant
USB Type - C specification
US10651612: Electrical connector with lower profile
USB Type - C specifications
USB (Universal Serial Bus) Type C
USB Type - C has
USB Type - C or
USB Type - C standard
USB Type - C to

Figure 5.5: Subgraph of USB type C related groups

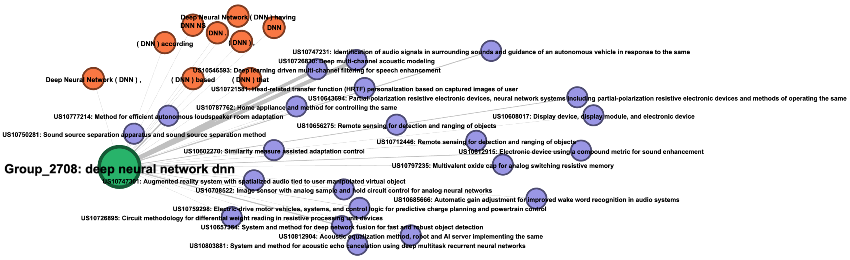

Deep Neural Network (DNN) having
DNN NS
DNN .
DNN
(DNN) according
(DNN) ,
US10747231: Identification of audio signals in surrounding sounds and guidance of an autonomous vehicle in response to the same
US10726830: Deep multi-channel acoustic modeling
Deep Neural Network (DNN) ,
(DNN) based
(DNN) that
US10546593: Deep learning driven multi-channel filtering for speech enhancement
US10721581: Head-related transfer function (HRTF) personalization based on captured images of user
US10643694: Partial-polarization resistive electronic devices, neural network systems including partial-polarization resistive electronic devices and methods of operating the same
US10787762: Home appliance and method for controlling the same
US10777214: Method for efficient autonomous loudspeaker room adaptation
US10608017: Display device, display module, and electronic device
US10656275: Remote sensing for detection and ranging of objects
US10750281: Sound source separation apparatus and sound source separation method
US10712446: Remote sensing for detection and ranging of objects
US10602270: Similarity measure assisted adaptation control
US10812915: Electronic device using a compound metric for sound enhancement
Group_2708: deep neural network dnn
US10797235: Multivalent oxide cap for analog switching resistive memory
US10747301: Augmented reality system with spatialized audio tied to user manipulated virtual object
US10708522: Image sensor with analog sample and hold circuit control for analog neural networks
US10685666: Automatic gain adjustment for improved wake word recognition in audio systems
US10759298: Electric-drive motor vehicles, systems, and control logic for predictive charge planning and powertrain control
US10726895: Circuit methodology for differential weight reading in resistive processing unit devices
US10657364: System and method for deep network fusion for fast and robust object detection
US10812904: Acoustic equalization method, robot and AI server implementing the same
US10803881: System and method for acoustic echo cancelation using deep multitask recurrent neural networks

Figure 5.6: Subgraph of DNN related groups

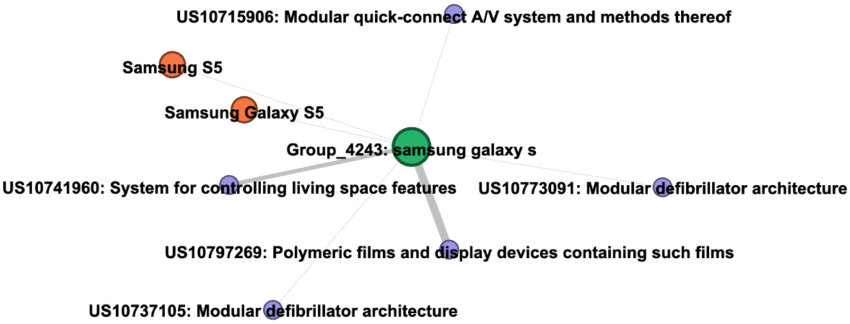

US10715906: Modular quick-connect A/V system and methods thereof
Samsung S5
Samsung Galaxy S5
Group_4243: samsung galaxy s
US10741960: System for controlling living space features
US10773091: Modular defibrillator architecture
US10797269: Polymeric films and display devices containing such films
US10737105: Modular defibrillator architecture

Figure 5.7: Subgraph of Samsung Galaxy related groups

5.5 Chapter summary

The knowledge graph has been recently attracting attention in the field of patent analysis as a useful tool to summarize and represent information from patent filings. Past research has mainly relied on extracting keywords to summarize and represent the information enclosed in patent filings. While keyword extraction models do deliver meaningful insights, named entities such as technological concepts, specific techniques used, name of the devices or end product, and the associated company names may additionally provide richer and deeper understanding of the innovations and technology underlying the patent filings.

In this study, we construct a semiconductor-related concept and entity knowledge graph by applying a novel edge updating neural network algorithm on patent claims. More specifically, our proposed model builds a knowledge network of semiconductor-related named entities from the patent filings. During this process, named entities with different surface forms, but of identical meanings, are placed into unique groups, hence providing a clearer picture and better understanding of the patent filings in hand. Our proposed model shows the highest performance on both the NEN and document retrieval tasks against that of standard baseline models. Further, experiment results show that the proposed knowledge graph construction method is robust to the out-of-vocabulary problem.

While the proposed model has showed great performances, there still is a room for further development. Currently, our research focuses only on the topics involving semiconductor devices. A focus switch to other fields may lead to a clearer understanding of a different area of innovations, while an extension to encompass a greater range of topics will help assemble a more complete picture of the recent technological advances in general. In addition, this study uses the edge updating neural network approach to discover the inter-connectivity among named entities, whereas their relationship to the patent documents is determined only through the simple co-occurrence. By utilizing the techniques proposed in this research, defining the relationship between the document and named entities is our next research topic.

Chapter 6

Conclusion

6.1 Contributions

Building the more complete named entity knowledge graph requires human power and costs. Our research aims to achieve the automation of more complete named entity knowledge graph. We propose dictionary Construction for Named Entity Normalization, named entity normalization model using edge weight updating neural network, and framework for building knowledge graph using named entity recognition and normalization models. It is expected to lower the barriers in the text mining field that may occur due to the absence of named entity normalization datasets and dictionaries. By suggesting the better performing named entity normalization model, our research will be helpful in text mining research in various financial and technical document analysis fields such as information retrieval, text classification, and sentiment analysis. In practice, using the named entity dictionary, named entity normalization model, and knowledge graph construction framework, it is possible to automate some of the tasks performed by humans, which is expected to bring improvement in business performance.

6.2 Future work

Our study presents the named entity normalization datasets from finance documents and patent documents. It is valuable to extend this to natural language processing. Our next research topic will focus on creating the named entity normalization dataset for the review texts. Furthermore, our knowledge graph construction framework can be applied to many other text mining studies. Conducting the studies on the applications of created knowledge graph will be beneficial to many practices. Also, this study uses the edge updating neural network approach to discover the inter-connectivity among named entities, whereas their relationship to the patent documents determined only through the simple co-occurrence. By utilizing the proposed techniques in this research, defining the relationship between the document and named entities is our next research topic.

Bibliography

[1] A. Abbas, L. Zhang, and S. U. Khan, *A literature review on the state-of-the-art in patent analysis*, World Patent Information, 37 (2014), pp. 3–13.

[2] M. Allahgholi, H. Rahmani, D. Javdani, Z. Sadeghi-Adl, A. Bender, D. Módos, and G. Weiss, *Ddrel: From drug-drug relationships to drug repurposing*, Intelligent Data Analysis, 26 (2022), pp. 221–237.

[3] D. Araci, *Finbert: Financial sentiment analysis with pre-trained language models*, arXiv preprint arXiv:1908.10063, (2019).

[4] A. R. Aronson, *Effective mapping of biomedical text to the umls metathesaurus: the metamap program.*, in Proceedings of the AMIA Symposium, American Medical Informatics Association, 2001, p. 17.

[5] A. Arratia, L. A. Belanche, and L. Fábregues, *An evaluation of equity premium prediction using multiple kernel learning with financial features*, Neural Processing Letters, (2019), pp. 1–18.

[6] I. Beltagy, K. Lo, and A. Cohan, *Scibert: A pretrained language model for scientific text*, arXiv preprint arXiv:1903.10676, (2019).

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information*, Transactions of the association for computational linguistics, 5 (2017), pp. 135–146.

[8] R. Bossy, L. Deléger, E. Chaix, M. Ba, and C. Nédellec, *Bacteria biotope at bionlp open shared tasks 2019*, in Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, 2019, pp. 121–131.

[9] H. Cho, W. Choi, and H. Lee, *A method for named entity normalization in biomedical articles: application to diseases and plants*, BMC bioinformatics, 18 (2017), p. 451.

[10] J. Choi and Y.-S. Hwang, *Patent keyword network analysis for improving technology development efficiency*, Technological Forecasting and Social Change, 83 (2014), pp. 170–182.

[11] B. S. Corba, E. Egrioglu, and A. Z. Dalar, *Ar–arch type artificial neural network for forecasting*, Neural Processing Letters, 51 (2020), pp. 819–836.

[12] A. P. Davis, C. G. Murphy, C. A. Saraceni-Richards, M. C. Rosenstein, T. C. Wiegers, and C. J. Mattingly, *Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks*, Nucleic acids research, 37 (2009), pp. D786–D792.

[13] D. Demner-Fushman, S. E. Shooshan, L. Rodriguez, A. R. Aronson, F. Lang, W. Rogers, K. Roberts, and J. Tonning, *A dataset of 200 structured product labels annotated for adverse drug reactions*, Scientific data, 5 (2018), p. 180001.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[15] H. Dhayne, R. Kilany, R. Haque, and Y. Taher, *Emr2vec: Bridging the gap between patient data and clinical trial*, Computers & Industrial Engineering, 156 (2021), p. 107236.

[16] R. I. Doğan, R. Leaman, and Z. Lu, *Ncbi disease corpus: a resource for disease name recognition and concept normalization*, Journal of biomedical informatics, 47 (2014), pp. 1–10.

[17] J. D'Souza and V. Ng, *Sieve-based entity linking for the biomedical domain*, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 297–302.

[18] S. Francis, J. Van Landeghem, and M.-F. Moens, *Transfer learning for named entity recognition in financial and biomedical documents*, Information, 10 (2019), p. 248.

[19] K. Fu, J. Cagan, K. Kotovsky, and K. Wood, *Discovering structure in design databases through functional and surface based mapping*, Journal of mechanical Design, 135 (2013), p. 031006.

[20] K. Fu, J. Chan, J. Cagan, K. Kotovsky, C. Schunn, and K. Wood, *The meaning of "near" and "far": the impact of structuring design databases*

and the effect of distance of analogy on design output, Journal of Mechanical Design, 135 (2013).

[21] M. Galkin, S. Auer, M.-E. Vidal, and S. Scerri, *Enterprise knowledge graphs: A semantic approach for knowledge management in the next generation of enterprise information systems.*, in ICEIS (2), 2017, pp. 88–98.

[22] O. Ghiasvand and R. J. Kate, *Uwm: Disorder mention extraction from clinical text using crfs and normalization using learned edit distance patterns.*, in SemEval@ COLING, 2014, pp. 828–832.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 27 (2014).

[24] A. Gupta, V. Dengre, H. A. Kheruwala, and M. Shah, *Comprehensive review of text-mining applications in finance*, Financial Innovation, 6 (2020), pp. 1–25.

[25] J. Hakenberg, M. Gerner, M. Haeussler, I. Solt, C. Plake, M. Schroeder, G. Gonzalez, G. Nenadic, and C. M. Bergman, *The gnat library for local and remote gene mention normalization*, Bioinformatics, 27 (2011), pp. 2769–2771.

[26] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck, *Prominer: rule-based protein and gene entity recognition*, BMC bioinformatics, 6 (2005), pp. 1–9.

[27] X. Hao, Z. Ji, X. Li, L. Yin, L. Liu, M. Sun, Q. Liu, and R. Yang, *Construction and application of a knowledge graph*, Remote Sensing, 13 (2021), p. 2511.

[28] A. Hermans, L. Beyer, and B. Leibe, *In defense of the triplet loss for person re-identification*, arXiv preprint arXiv:1703.07737, (2017).

[29] E. Hoffer and N. Ailon, *Deep metric learning using triplet network*, in International workshop on similarity-based pattern recognition, Springer, 2015, pp. 84–92.

[30] S. H. Jeon and S. Cho, *Edge weight updating neural network for named entity normalization*, Neural Processing Letters, (2022), pp. 1–22.

[31] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, *A survey on knowledge graphs: Representation, acquisition, and applications*, IEEE Transactions on Neural Networks and Learning Systems, 33 (2021), pp. 494–514.

[32] Z. Ji, Q. Wei, and H. Xu, *Bert-based ranking for biomedical entity normalization*, AMIA Summits on Translational Science Proceedings, 2020 (2020), p. 269.

[33] V. Jijkoun, M. A. Khalid, M. Marx, and M. De Rijke, *Named entity normalization in user generated content*, in Proceedings of the second workshop on Analytics for noisy unstructured text data, 2008, pp. 23–30.

[34] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, *Fasttext.zip: Compressing text classification models*, arXiv preprint arXiv:1612.03651, (2016).

[35] N. Kang, B. Singh, Z. Afzal, E. M. van Mulligen, and J. A. Kors, *Using rule-based natural language processing to improve disease normalization in biomedical text*, Journal of the American Medical Informatics Association, 20 (2013), pp. 876–881.

[36] I. Karadeniz and A. Özgür, *Linking entities through an ontology using word embeddings and syntactic re-ranking*, BMC bioinformatics, 20 (2019), pp. 1–12.

[37] J. Kim, T. Kim, S. Kim, and C. D. Yoo, *Edge-labeling graph neural network for few-shot learning*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11–20.

[38] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, *Overview of bionlp'09 shared task on event extraction*, in Proceedings of the BioNLP 2009 workshop companion volume for shared task, 2009, pp. 1–9.

[39] L. Kim, E. Yahia, F. Segonds, P. Véron, and A. Mallet, *i-dataquest: A heterogeneous information retrieval tool using data graph for the manufacturing industry*, Computers in Industry, 132 (2021), p. 103527.

[40] Y. G. Kim, J. H. Suh, and S. C. Park, *Visualization of patent analysis for emerging technology*, Expert systems with applications, 34 (2008), pp. 1804–1812.

[41] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[42] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, arXiv preprint arXiv:1609.02907, (2016).

[43] R. Klinger, C. Kolářik, J. Fluck, M. Hofmann-Apitius, and C. M. Friedrich, *Detection of iupac and iupac-like chemical names*, Bioinformatics, 24 (2008), pp. i268–i276.

[44] C. Kolárik, R. Klinger, C. M. Friedrich, M. Hofmann-Apitius, and J. Fluck, *Chemical names: terminological resources and corpora annotation*, in Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference), 2008.

[45] E. Krivosheev, M. Atzeni, K. Mirylenka, P. Scotton, and F. Casati, *Siamese graph neural networks for data integration*, arXiv preprint arXiv:2001.06543, (2020).

[46] S. Kullback and R. A. Leibler, *On information and sufficiency*, The annals of mathematical statistics, 22 (1951), pp. 79–86.

[47] R. Leaman, R. Islamaj Doğan, and Z. Lu, *Dnorm: disease name normalization with pairwise learning to rank*, Bioinformatics, 29 (2013), pp. 2909–2917.

[48] R. Leaman and Z. Lu, *Taggerone: joint named entity recognition and normalization with semi-markov models*, Bioinformatics, 32 (2016), pp. 2839–2846.

[49] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, *Biobert: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics, 36 (2020), pp. 1234–1240.

[50] J.-S. Lee and J. Hsiang, *Patentbert: Patent classification with fine-tuning a pre-trained bert model*, arXiv preprint arXiv:1906.02124, (2019).

[51] V. I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, in Soviet physics doklady, vol. 10, 1966, pp. 707–710.

[52] H. Li, Q. Chen, B. Tang, X. Wang, H. Xu, B. Wang, and D. Huang, *Cnn-based ranking for biomedical entity normalization*, BMC bioinformatics, 18 (2017), pp. 79–86.

[53] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, *Biocreative v cdr task corpus: a resource for chemical disease relation extraction*, Database, 2016 (2016).

[54] L. Li, P. Wang, J. Yan, Y. Wang, S. Li, J. Jiang, Z. Sun, B. Tang, T.-H. Chang, S. Wang, et al., *Real-world data medical knowledge graph: construction and applications*, Artificial intelligence in medicine, 103 (2020), p. 101817.

[55] B. Liu, T. Zhang, D. Niu, J. Lin, K. Lai, and Y. Xu, *Matching long text documents via graph convolutional networks*, arXiv preprint arXiv:1802.07459, (2018), pp. 2793–2799.

[56] Y. Liu, F. Gu, Y. Wu, X. Gu, and J. Guo, *A metrics-based meta-learning model with meta-pretraining for industrial knowledge graph construction*, Computers in Industry, 143 (2022), p. 103753.

[57] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, arXiv preprint arXiv:1711.05101, (2017).

[58] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[59] I. Mondal, S. Purkayastha, S. Sarkar, P. Goyal, J. Pillai, A. Bhattacharyya, and M. Gattu, *Medical entity linking using triplet network*, arXiv preprint arXiv:2012.11164, (2020).

[60] J. Mueller and A. Thyagarajan, *Siamese recurrent architectures for learning sentence similarity*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 2016.

[61] I. O. Mulang', K. Singh, C. Prabhu, A. Nadgeri, J. Hoffart, and J. Lehmann, *Evaluating the impact of knowledge graph context on entity disambiguation models*, in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2157–2160.

[62] P. Neculoiu, M. Versteegh, and M. Rotaru, *Learning text similarity with siamese recurrent networks*, in Proceedings of the 1st Workshop on Representation Learning for NLP, 2016, pp. 148–157.

[63] J. Niu, Y. Yang, S. Zhang, Z. Sun, and W. Zhang, *Multi-task character-level attentional networks for medical concept normalization*, Neural Processing Letters, 49 (2019), pp. 1239–1256.

[64] H. Noh, Y. Jo, and S. Lee, *Keyword selection and processing strategy for applying text mining to patent analysis*, Expert Systems with Applications, 42 (2015), pp. 4348–4360.

[65] J. Pennington, R. Socher, and C. D. Manning, *Glove: Global vectors for word representation*, in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[66] M. C. Phan, A. Sun, and Y. Tay, *Robust representation learning of biomedical names*, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3275–3285.

[67] H. Rahmani, H. Blockeel, and A. Bender, *Using a human disease network for augmenting prior knowledge about diseases*, Intelligent Data Analysis, 19 (2015), pp. 897–916.

[68] ——, *Using a human drug network for generating novel hypotheses about drugs*, Intelligent Data Analysis, 20 (2016), pp. 183–197.

[69] H. Rahmani, B. Ranjbar-Sahraei, G. Weiss, and K. Tuyls, *Entity resolution in disjoint graphs: an application on genealogical data*, Intelligent Data Analysis, 20 (2016), pp. 455–475.

[70] L. Ramos, *Semantic web for manufacturing, trends and open issues: Toward a state of the art*, Computers & Industrial Engineering, 90 (2015), pp. 444–460.

[71] T. Ranasinghe, C. Orasan, and R. Mitkov, *Semantic textual similarity with siamese neural networks*, in Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), 2019, pp. 1004–1011.

[72] T. Rocktäschel, M. Weidlich, and U. Leser, *Chemspot: a hybrid system for chemical named entity recognition*, Bioinformatics, 28 (2012), pp. 1633–1640.

[73] A. Rosenberg and J. Hirschberg, *V-measure: A conditional entropy-based external cluster evaluation measure*, in Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), 2007, pp. 410–420.

[74] E. F. Sang and F. De Meulder, *Introduction to the conll-2003 shared task: Language-independent named entity recognition*, arXiv preprint cs/0306050, (2003).

[75] S. Sarica, J. Luo, and K. L. Wood, *Technet: Technology semantic network based on patent data*, Expert Systems with Applications, 142 (2020), p. 112995.

[76] F. Schroff, D. Kalenichenko, and J. Philbin, *Facenet: A unified embedding for face recognition and clustering*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[77] M. Schuster and K. K. Paliwal, *Bidirectional recurrent neural networks*, IEEE transactions on Signal Processing, 45 (1997), pp. 2673–2681.

[78] W. Seo, *A patent-based approach to identifying potential technology opportunities realizable from a firm's internal capabilities*, Computers & Industrial Engineering, 171 (2022), p. 108395.

[79] Y. Si, J. Wang, H. Xu, and K. Roberts, *Enhancing clinical concept extraction with contextual embeddings*, Journal of the American Medical Informatics Association, 26 (2019), pp. 1297–1304.

[80] L. Smith, L. K. Tanabe, R. J. nee Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, et al.,

Overview of biocreative ii gene mention recognition, Genome biology, 9 (2008), p. S2.

[81] C. Sun, L. Lin, M. Liu, B. Liu, and X. Sha, *A product named entity normalization method based on entity relations*, in 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012), vol. 1, IEEE, 2012, pp. 166–169.

[82] Y. Sun, W. Liu, G. Cao, Q. Peng, J. Gu, and J. Fu, *Effective design knowledge abstraction from chinese patents based on a meta-model of the patent design knowledge graph*, Computers in Industry, 142 (2022), p. 103749.

[83] M. Sung, H. Jeon, J. Lee, and J. Kang, *Biomedical entity representations with synonym marginalization*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3641–3650.

[84] H. Suominen, S. Salanterä, S. Velupillai, W. W. Chapman, G. Savova, N. Elhadad, S. Pradhan, B. R. South, D. L. Mowery, G. J. Jones, et al., *Overview of the share/clef ehealth evaluation lab 2013*, in International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2013, pp. 212–231.

[85] Z. Wang, C.-H. Chen, P. Zheng, X. Li, and W. Song, *A hypergraph-based approach for context-aware smart product-service system configuration*, Computers & Industrial Engineering, 163 (2022), p. 107816.

[86] C.-H. Wei and H.-Y. Kao, *Cross-species gene normalization by species inference*, BMC bioinformatics, 12 (2011), p. S5.

[87] L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, G. Ceder, and A. Jain, *Named entity recognition and normalization applied to large-scale information extraction from the materials science literature*, Journal of chemical information and modeling, 59 (2019), pp. 3692–3702.

[88] D. Wright, *NormCo: Deep disease normalization for biomedical knowledge base construction*, PhD thesis, UC San Diego, 2019.

[89] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., *Google's neural machine translation system: Bridging the gap between human and machine translation*, arXiv preprint arXiv:1609.08144, (2016).

[90] Z. Xu and Y. Dang, *Solution knowledge mining and recommendation for quality problem-solving*, Computers & Industrial Engineering, 159 (2021), p. 107313.

[91] B. Yoon and Y. Park, *A text-mining-based patent network: Analytical tool for high-technology trend*, The Journal of High Technology Management Research, 15 (2004), pp. 37–50.

[92] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai, *Hard-aware point-to-set deep metric for person re-identification*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 188–204.

국문초록

텍스트 마이닝은 다양한 인사이트를 얻기 위해 문서에서 정보를 추출하는 것을 목표로 한다. 문서의 정보를 표현하는 방식 중 하나인 지식 그래프는 다양한 문서에서 더욱 풍부한 정보를 제공한다. 기존 연구들은 텍스트 마이닝 기법을 이용하여 문서의 정보들로 기술 트리 또는 개념 네트워크를 구축하거나 키워드 및 구문을 추출하였다. 본 논문에서는 고유명사를 이용하여 지식 그래프를 구축하기 위한 프레임워크를 제안한다. 본 논문의 지식 그래프 구축 프레임워크는 다음과 같은 조건을 만족한다. (1) 고유명사를 사람이 이해하기 쉬운 형태로 추출한다. (2) 기존 고유명사 정규화 연구가 활발했던 생물정보학 외에 금융 문서, 반도체 관련 특허 문서에서 추출한 고유명사로 고유명사 정규화 데이터셋을 구축한다. (3) 더 나은 성능의 고유명사 정규화 모델을 구축한다. (4) 다양한 형태의 동일한 의미를 가진 고유명사를 그룹화하여 지식 그래프를 구축한다.

주요어: 고유명사 정규화, 간선 가중치 갱신 인공 신경망, 고유명사 지식 그래프, 키워드 추출

학번: 2016-21122

감사의 글

그동안 본 논문이 완성되기까지 학문적인 지도와 다양한 산학 협력 과제 기회를 주셔서 제가 더욱 성장할 수 있게 지도해 주신 조성준 교수님께 감사의 인사를 올립니다.

본 논문을 심사해 주신 이재욱 교수님, 백복현 교수님, 이영훈 교수님, 고태훈 교수님께 감사드립니다. 또한, 다양한 수업을 통해 학문적 깊이를 더 할 수 있게 지도해주신 모든 교수님께 감사함을 전하고 싶습니다.

연구실에서도 함께 수업을 수강하고, 프로젝트를 진행하고, 논문 작성을 함께한 연구실 선배님, 후배님 그리고 동기분들께도 진심으로 감사합니다.