WEAKLY-SUPERVISED RELATION EXTRACTION

ZAYIF DENETLENEN İLİŞKİ ÇIKARIMI

SEÇKİN ŞEN

PROF. DR. İLYAS ÇİÇEKLİ
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

September 2023

ABSTRACT

WEAKLY-SUPERVISED RELATION EXTRACTION

Seçkin Şen

Master of Science, Computer Engineering

Supervisor: Prof. Dr. İlyas Çiçekli

September 2023, 85 pages

Relation extraction is a crucial element for numerous natural language processing applications, including text summarization and question answering. It is noteworthy that there are diverse methodologies for relation extraction, and the majority of them adopt the supervised learning approach, which necessitates a substantial training dataset. These extensive datasets must be hand-labeled by experts, making the annotation process time-consuming and expensive. Another approach that is utilized in this thesis is called weak supervised relation extraction. Using weak supervised learning, the cost of training data labeling can be reduced. In this thesis, we propose a weakly supervised relation extraction approach that is inspired by another weakly supervised model named REPEL. Both in REPEL and our relation extraction approach, extraction patterns are derived from unlabeled texts using given relation seed examples. In order to extract more useful extraction patterns, we introduce the use of labeling functions in our method. These labeling functions consist of simple rules to analyze the candidate pattern's syntax and these labeling functions help to extract more confident candidate patterns. Our proposed method tests on the same dataset used by REPEL in order to compare our results with the results obtained by REPEL. Tests are conducted both in English and Turkish. Both systems require a number of relation

seed examples for learning patterns from the unlabeled data. When fewer relation seed examples are used our method outperforms REPEL significantly. In experimental tests, our approach generally gives better results than REPEL for both languages. For the English test, approximately 15 times more successful than REPEL with few relation seeds. Even with more relation seeds, our approach remains more successful.

Keywords: weakly supervised learning, relation extraction, information extraction, natural language processing, data programming

ÖZET

ZAYIF DENETLENEN İLİŞKİ ÇIKARIMI

Seçkin Şen

Yüksek Lisans, Bilgisayar Mühendisliği

Danışman: Prof. Dr. İlyas Çiçekli

Haziran 2023, 85 sayfa

İlişki çıkarımı, soru yanıtlama ve metin özetleme gibi birçok doğal dil işleme uygulaması için çok önemlidir. İlişki çıkarımı için farklı yaklaşımlar olmasına rağmen bunların çoğu, geniş bir eğitim veri seti gerektiren denetimli öğrenme yaklaşımını kullanır. Bu kapsamlı veri kümelerinin uzmanlar tarafından elle etiketlenmesi gerekir, bu da eğitim sürecini zaman alıcı ve pahalı hale getirir. Bu tezde kullanılan yaklaşıma zayıf denetimli ilişki çıkarımı adı verilmektedir. Zayıf denetimli öğrenme kullanılarak eğitim verileri etiketlemenin maliyeti azaltılabilir. Bu tezde, zayıf denetlenen başka bir model olan REPEL'den ilham alan, zayıf denetlenen bir ilişki çıkarma yaklaşımı öneriyoruz. Hem REPEL'de hem de ilişki çıkarma yaklaşımımızda, başta verilen ilişki örnekleri kullanılarak etiketlenmemiş metinlerden yeni çıkarım örüntüleri türetilir. Daha kullanışlı çıkarım örüntüleri elde etmek için yöntemimizde etiketleme fonksiyonlarının kullanımını tanıtıyoruz. Bu etiketleme işlevleri, aday örüntünün sözdizimini analiz etmeye yönelik basit kurallardan oluşur ve bu etiketleme işlevleri, daha güvenli aday kalıplarının çıkarılmasına yardımcı olur. Hem İngilizce hem de Türkçe olarak yapılan testlerde REPEL ve bu tezde önerilen model aynı veri seti ile eğitilerek sonuçlar karşılaştırılmıştır. Her iki sistem de etiketlenmemiş verilerden örnekleri öğrenmek için bir dizi ilişki örneğine ihtiyaç duyar. Daha az ilişki örneği kullanıldığında yöntemimiz

REPEL'den önemli ölçüde daha iyi performans gösterir. Deneysel testlerde yaklaşımımız genellikle her iki dil için de REPEL'den daha iyi sonuçlar verir. İngilizce testi için, az sayıda ilişki örneği ile önerilen model olan REPEL'den yaklaşık 15 kat daha başarılıdır. Daha fazla ilişki tohumu olsa bile yaklaşımımız daha başarılı olmaya devam etmektedir.

Keywords: zayıf denetimli öğrenme, ilişki çıkarma, bilgi çıkarma, doğal dil işleme, veri programlama

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. İlyas ÇİÇEKLİ for guiding me during my thesis study. I cannot express my gratitude for his invaluable patience and feedback.

I am also grateful to my beloved parents and brother for always encouraging me, even when I lost faith in myself and decided to give up. As with all my achievements in my life, I dedicate this thesis to them for their endless love and support.

Lastly, I would like to thank my friends and coworkers for their support and help with me.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

AI	:	**A**rtificial **I**ntelligence
IE	:	**I**nformation **E**xtraction
KBC	:	**K**nowledge **B**ase **C**ompletion
NE	:	**N**amed **E**ntity
NEL	:	**N**amed **E**ntity **L**inking
NER	:	**N**amed **E**ntity **R**ecognition
NLP	:	**N**atural **L**anguage **P**rocessing
OCR	:	**O**ptical **C**haracter **R**ecognition
POS	:	**P**art **O**f **S**peech
RE	:	**R**elation **E**xtraction
SME	:	**S**ubject **M**atter **E**xpert

1. INTRODUCTION

With the development of algorithms for artificial intelligence (AI), the laborious, dangerous and complex tasks that humans had to deal with have been transferred to robots and software. It provides systems that make fewer mistakes, are faster and work perpetually have been created. One of the areas where AI is used is information extraction (IE) from texts. Most of the texts used in daily life do not have a certain structure. Emails, social media posts and chat messages may differ from each other in terms of content or length. The process of extracting the relation information in these texts is called relation extraction (RE) and falls within the field of natural language processing (NLP), one of the sub-branches of AI.

AI algorithms have been developing very rapidly in nowadays. One of the most important reasons is the incredible increase in the amount of data. With the increase in data, AI algorithms that are trained with data have started to give very good results. However, these algorithms require a large amount of labeled data to be trained. Data labeling needs to be done by humans and takes time. Sometimes the data to be tagged may require expertise in a particular field, and it can be costly for an expert to dedicate their time to tagging data. Even if time and cost are not a problem, an expert may not be available to label the data. Another problem is that too much data is produced. There is no way to manually label all data. These problems also apply to relational extraction. Weak supervision has emerged as a solution for such situations. With weak learning, the process of labeling the data can be automated. It is possible to self-train the model by utilizing a small amount of tagged ground truth data. Although the accuracy of the data labeled by weak learning is not as high as that of the data labeled by humans, the gap between the accuracy of the trained models narrows as the dataset grows [3].

1.1. Scope Of The Thesis

In this study, a weak supervised relation extraction model is proposed. The model tries to extract relationship information between the named entities in the sentence. By analyzing a

few examples called seeds at the beginning, other examples of the same relationship type are tried to be extracted from the text. Sentences are examined from three different perspectives.

The first perspective is syntactic examination. The relationship between entities is analyzed syntactically by subtracting the shortest dependency path between entities in the sentence. This shortest path is called a pattern and the rest of the text is examined at how many instances there are in the text of this pattern and how many of these examples are the same as the instances in the seed. The larger the number of common samples, the more likely this candidate pattern is to be a significant pattern. Hence, extracted instances of the pattern that are not in the seed document can be assumed as correct samples if other extracted instances of the pattern are in the seed.

The second perspective is a semantic examiner. It checks how similar meaning to each other the named entity (NE) pair that is extracted. The closer the entities, the higher the accuracy of the pattern found. Because interrelated entities are assumed to be semantically similar.

Lastly, the proposed model has a lexical examiner. Not only named entities but the rest words of the pattern are examined. It is checked the pattern has a point to sign that a relation such as a specific word or known NE attribute. Since it is of unstructured nature, it is not expected that every pattern will have the same signs. Therefore, although the examined cases do not give meaningful results for all patterns, they can give information about whether some patterns contain a relationship. For example, in sentences with a ”capitalOf” relation, the probability of using the word ”capital” is higher than other relations. Another example is checking if named entities represent two ”PERSON” with the same last name. If looking for the ”childOf” relationship, this information is very important, while it is not important for the ”capitalOf” relationship. Just like in the entity examples, the words to be used for labeling functions can be written in the seed. If any of these words appear in the pattern, the pattern is likely to contain a relationship.

1.2. Contributions

The algorithm used in this study was inspired by the REPEL[2] study. REPEL has syntactic and semantic modules mentioned above for relation extraction, but the lexical module is intrinsic to the proposed model. The effect of the newly added module was tried to be observed by comparing the developed model with REPEL. Models were compared using relation extraction and knowledge base completion (KBC) tests. Tests are conducted in English and Turkish. With these tests, it was tried to show the effect of adding a new module and editing others. Tests are conducted by using a dataset that has four different types of relationships in each sentence. Each sentence has one of four types. Models trained with the same train dataset and having the same seed NE samples were expected to detect these relationships and classify them correctly. In the KBC test, the models trained with the same data set and having the same seed NE samples were expected to match the capitals and countries correctly. When the results are examined, the newly added module has a visible contribution. Especially as the number of seed NE pairs is reduced, the difference gets bigger.

In the present study, we address the aforementioned inadequacies by presenting a novel methodology. The primary inclusions of this manuscript are succinctly delineated below.

- A weakly supervised relation extraction model has been proposed and it greatly reduces the need for labeled data.
- Unlike most of the previous works, we used Weak supervision and data programming together, reducing the dependency of weak supervision on labeled data.
- The results of our evaluation demonstrate that the model presented in this study yields superior outcomes in comparison to the model used for comparison.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 elaborates on the impetus behind our research, the contributions that we aim to make, and the scope of our dissertation.
- Chapter 2 furnishes fundamental knowledge concerning utilized algorithms and concepts.
- Chapter 3 gives a summary of related works.
- Chapter 4 introduces the proposed model and differences from REPEL.
- Chapter 5 demonstrates the results of the conducted experiments.
- Chapter 6 presents a comprehensive overview of the thesis and explores potential avenues for future research.

2. BACKGROUND OVERVIEW

In order to facilitate comprehension, we present fundamental contextual information in six distinct sections: weak supervision, information extraction, relation extraction, data programming, dependency parsing, and text vectorization.

2.1. Weak Supervision

Weak learning is the process of predicting the label programmatically using different methods instead of labeling the data manually. It first appeared in the field of biomedicine [4]. Labels obtained with weak learning may not be as accurate (inexact) or detailed (incomplete) as labeled by the Subject Matter Expert (SME). One reason why these programmatically created tags are called weak is that the error margin of the created labels is higher than the SME labels. These noisy labels are called weak labels. Another reason is that information obtained as a result of weak learning may not be used directly for labeling [5]. Weak learning labeling process increases the accuracy rate as the amount of data increases. In fact, in the studies carried out by Robinson et. al [3] showed that models using weak supervised learning with little ground-truth data have learning ability and can learn faster than models using traditional ground-truth datasets. Weakly supervised learning is divided into three main categories according to the problem it deals with. Figure 2.1 shows the classification of weak supervision and they are explained in the following subsections. The proposed model uses semi-supervised learning and transfer learning approaches together. By using these approaches together, it is aimed to learn better than simple models. Details are given in Chapter 3.

2.1.1. Incomplete Supervision

Only a fraction of the dataset has received annotation, indicating that a significant proportion remains unannotated. The remaining samples do not have labels. The weak supervision method dealing with this problem is called Incomplete Supervision. It is divided into three

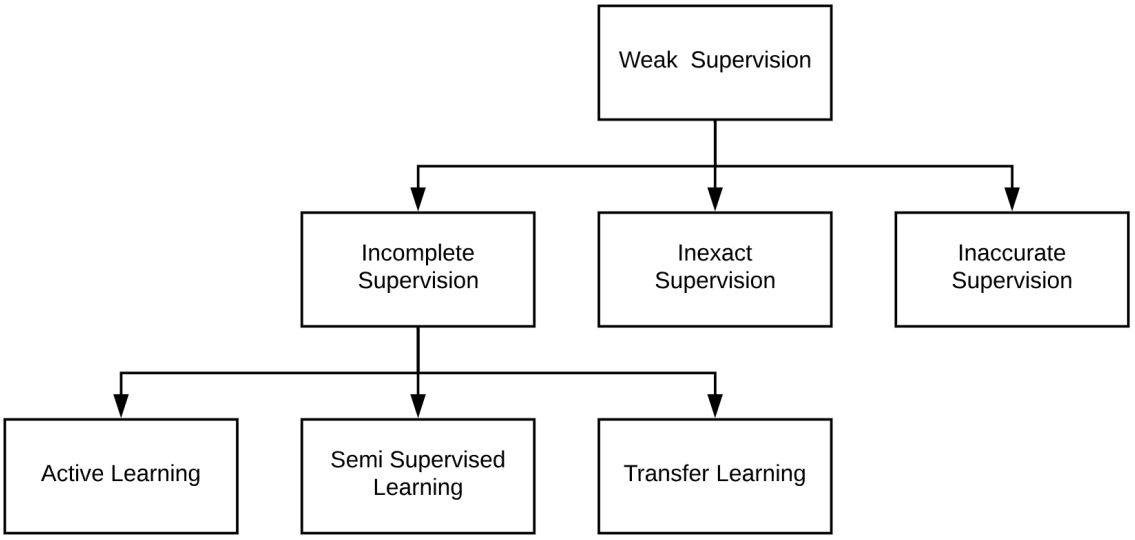

Figure 2.1 Classification of Weak Supervision

sub-headings according to the approaches used. They are active learning, semi-supervised learning and transfer learning.

2.1.1.1. Active Learning Data is labeled with assistance from SME. The cost of tagging is simply assumed to be the number of queries to the SME and the aim of this approach is to reduce this cost as much as possible [6]. For this, only a very small part of the unlabeled dataset is shown to the SME, and the model automatically labels the rest of the dataset according to the labels that come from SME. Two different criteria come to the fore regarding how to select the data to be sent for the query and they are called informativeness and representativeness [7]. In representativeness, it is inspected how much the labeled data reflects the structure of the dataset [8]. In informativeness, it is checked how much uncertainty will be removed when the selected data is labeled. Informativeness is also divided into uncertainty sampling and query-by-committee approaches. In uncertainty sampling, a solitary learner is employed, whereby the learner's least confident instance is labeled and subsequently forwarded to the SME. On the contrary, in the alternative method, multiple learners are trained, and the instance that elicits the most disagreement is referred to the SME. In order to address the cluster structure of unlabeled data, informativeness approaches

are generally employed[9, 10]. The common weakness of informativeness methods is that the tags are too dependent on the data. The performance of the model may be unstable, especially if there are few labeled data. The general weakness of the representativeness approach is that it is too dependent on unlabeled data while creating the cluster architecture. In current studies, a balanced use of these two approaches is studied [7, 11].

2.1.1.2. Semi-Supervised Learning The model labels data itself by using the information from the feature vector of the data without the help of SME. There are two basic assumptions about the distribution of data in semi-supervised learning. According to the manifold assumption, data with similar properties receive similar labels. Under the cluster assumption, the data is considered to be part of the clusters. Therefore, data in the same cluster have similar properties and receive the same label. Both assumptions say that unlabeled data can be labeled according to nearby labels.

Semi-supervised learning can be classified into two distinct methods based on the test dataset utilized. The closed-world approach is implemented in transductive learning, wherein the test data is chosen from the set of unlabeled data. The model anticipates the test dataset, and its objective is to optimize its performance on this set. Conversely, in pure semi-supervised learning, the test dataset is not selected from the unlabeled data, but rather a separate test dataset is utilized.

The approaches used for semi-supervised learning can be discussed under four headings.

- **Generative Methods:** Labeled and unlabeled data are posited to originate from identical structures, with the labels of the unlabeled data being construed as absent parameters of the said structures [12].
- **Graph-based Methods:** A graph is created where the nodes are the train data and the edges express the proximity and similarity between the nodes. According to some criteria such as minimum cut, tag information is spread throughout the graph [13]. The success rate in this method is dependent on the graph created [14, 15]. This approach

usually introduces $\mathcal{O}(n^2)$ memory and almost $\mathcal{O}(n^3)$ computational complexity. There are problems with the approach in terms of scalability. Graph-based methods have a transductive learning structure. It is difficult to classify new instances that are not in the training set and cannot be transferred to the graph [8].

- **Low-density Separation Methods:** This approach tries to keep the areas of the classes in the Input space as wide as possible and put the borders in low density regions. One of the most well-known methods is S3VM (semi-supervised support vector machine). It keeps the margin between classes smaller while correctly classifying data compared to normal SVMs. Since this method requires a lot of optimizations, studies on this method generally focus on optimization studies [16–18].

- **Disagreement-based Methods:** Multiple learners are trained and run together to label unlabeled data. Models of this method work in iterations. The most stable samples are selected in each iteration and get labeled. Learners can be further developed by combining them and making them an ensemble [19, 20].

2.1.1.3. Transfer Learning Frequently, models need external information beyond the dataset to understand the problem and language they are dealing with. Transfer learning is the transfer of information from a well-trained model that was previously used for another purpose to the new model in the initial phase. In cases where there is little labeled data or there is not enough training data, transfer learning can help train the model.

2.1.2. Inexact Supervision

All instances in the dataset have labels, but these labels are not detailed as much as desired. For example, in a dataset, all images have a coarse-grained label. However, objects found in these images are unlabeled. Weak supervision can be used to label objects such as cats, dogs or cars. Multi-instance learning is used for inexact supervision [21]. In the data set about which there is incomplete information, the samples are grouped in sequences called bags. Instances in the same bag are considered similar even though they are different from

each other. Labels are given for an entire bag instead of giving individual instances in the bag. The existence of positive and negative bags is a fundamental characteristic of multi-instance learning and labeled data. A bag is deemed positive when there exists at least one instance within the bag that is positively labeled. Conversely, if all instances within a bag are negatively labeled, the bag is deemed negative. This process constitutes the classification of imperceptible instances via tagged bags and data.

2.1.3. Inaccurate Supervision

Some of the instances in the dataset have incorrect labels. These need to be identified and corrected. To achieve this, the noise in the labels must first be detected [22]. The data-editing approach involves the creation of a relative neighborhood graph. Each node of the graph represents a training instance. In this graph, the links connecting nodes with different labels are called cutting-edge. If a node has too many cutting edges, the label of the node can be changed or deleted [23]. Crowdsourcing is another method used for inaccurate supervision [24].

2.2. Information Extraction

Written sources are frequently used in daily life. School, workplace or social media can be given as examples of places where texts are used frequently. With these texts, information can be stored and this information can be transferred between people. The information transferred to the texts can be very diverse. An entity, event, or relationship can be mentioned. Some texts have a regular organization and it is known what data will be found. For example, when the database tables are examined, the information is organized according to a format. Such texts are called structured texts. However, most texts do not have a regular structure. Texts in which what kind of information cannot be known beforehand are called unstructured texts. Most texts in daily life are unstructured. The process of discovering the information contained in the texts is called IE. IE is one of the sub-topics of NLP. Search engines, chatbots, for knowledge base construction, question answering systems, optical character

recognition (OCR) and social media algorithms can be given as examples of the daily use of IE [25, 26]. In structured texts, this can be done easily using regular expression, but more complex solutions may be required for unstructured texts. According to the sources used, we can examine IE under four main headings [27]. These resources are text, image, video and audio. The titles outside the scope of the thesis are not included. Figure 2.2 shows classification of text IE.

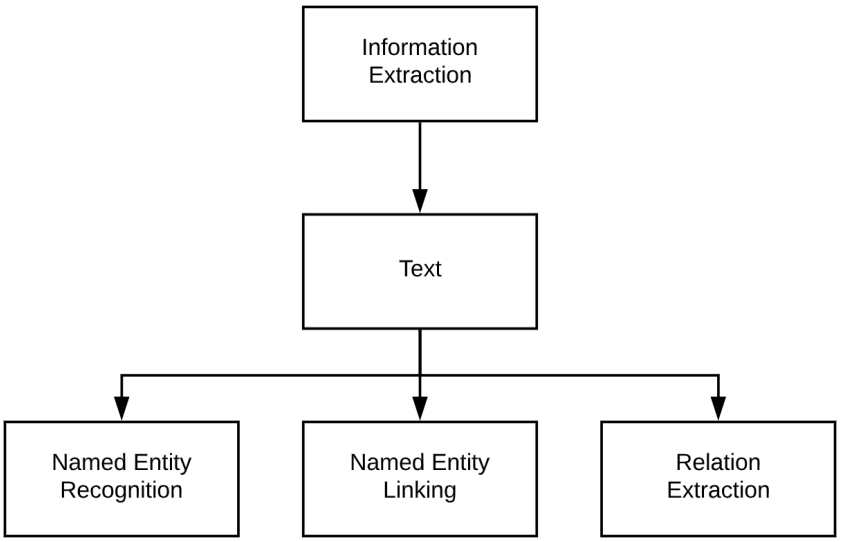

Figure 2.2 Classification of Text Information Extraction

2.2.1. Named Entity Recognition

In the field of Information Extraction (IE), specifically within the sub-branch of Named Entity Recognition (NER), the primary objective is to identify the entities present within a given text. These entities can manifest in various forms, including but not limited to individuals, locations, occurrences, monetary units, and dates. For instance, within the sentence, ”Ankara is the capital of Turkey,” both ”Ankara” and ”Turkey” represent named entities due to their classification as entities. Moreover, it is crucial to note that entities may possess distinct classes. PERSON, ORGANIZATION, LOCATION are some of the most used classes [28]. More detailed classes can also be created depending on the problem being dealt with. For instance, assets can be classified as FILM-TITLE or BOOK-TITLE instead

of TITLE. NER is one of the most studied NLP topics, is also frequently used in daily life. Translation applications can be given as a daily life use case of NER.

2.2.2. Named Entity Linking

All named entities discovered in the text may not be referring to different entities. An entity can have more than one name. In this case, named entities denoting the same entities must be linked. This process is called Named Entity Linking (NEL) [26]. For example, Bill Clinton and William Clinton are two different named entities representing the same person. Because they represent the same entity, they can be mapped to the same entity.

2.3. Relation Extraction

Some of the information in the text can be easily identified. For example, if a phone number can be determined using regular expressions. However, some information may not have a specific pattern. An example of this is the relationships between named entities. Trying to find the relationship between entities is called relation extraction (RE). NER and NEL are prerequisite tasks for RE. Named entities must be determined first to find the relationship between named entities. NEL, on the other hand, is not an essential step, but since it will reduce the number of named entities, it facilitates the finding of relationships between named entities.

Named entities and their inter-entity associations are sought to be identified through the use of RE. Nonetheless, it is not imperative for each Named Entity in the sentence to have a relationship. Thus, it becomes imperative to determine the association between the different entities.

Relation extraction is segregated into binary or high-order relationships based on the nature of the relationship at hand [29].In the realm of relation extraction, binary RE endeavors to ascertain the connection between two distinct entities, whereas high-order RE is aimed at identifying the relationship between multiple entities. Within the confines of this thesis, our

proposed model is focused on binary RE, and specifically, the examination of the relationship between two entities.

Relation extraction is divided into two subheadings according to the texts it receives as input. Relation extraction at sentence level and document level. In sentence level RE, the relationship is tried to be determined by only looking at the entities in the selected sentence. There can be duplicate NE pairs if there are more than one sentences that have the same named entities. Because every sentence is examined independently, the model needs to memorize too many patterns. Document level RE tries to make a relation extraction according to the information collected from the whole text. Named entities are extracted, and then the relation between named entities is tried to detect by detecting multiple sentences. Our proposed model extracts relations on sentence level. Sentences are considered independent of each other and patterns are extracted from sentences separately.

The extraction of relations between named entities can be achieved through the utilization of four distinct approaches in the field of Relation Extraction methods. These approaches include Supervised Learning, Unsupervised Learning, Semi Supervised Learning, and Weakly Supervised Learning. In the present thesis, the proposed method employed the Weakly Supervised Learning approach for the purpose of relation extraction.

2.3.1. Supervised Learning

Performing relation extraction using a model which is trained with labeled data is called supervised learning. Training data includes NE pairs and relations between them. The supervised model is trained with positive and negative samples (e.g. NONE is written on the label if unrelated samples are given) in the training dataset. Supervised learning approaches the relation extraction problem as a multiclass classification problem and tries to put the test data in the correct relation class. In general, it has two subheadings. In feature-based approaches, a feature vector is created by extracting lexical, semantic and syntactic features from the sample during training. Classification of test data is decided according to feature vectors. By examining named entities and other words between these entities, it is tried to

extract feature vector from these words. In feature-based approaches, it is necessary to extract a feature vector for each instance. Kernel-based approaches do not require preprocessing as in feature-based approaches. Using kernel functions, the similarity between two instances is measured and instances are classified using SVM.

2.3.2. Unsupervised Learning

In unsupervised learning, models label test dataset instances without any training. Clusters are created by looking at context similarity in clustering-based approaches. The created clusters are assigned a semantic relation name [30]. Although these models do not require training, the results are often inconclusive [31].

2.3.3. Semi Supervised Learning

A portion of the available dataset is labeled. The model tries to label the remaining parts by looking at the labels it has [29]. The process of labeling untagged instances can be thought of as propagating knowledge. The model labels instances according to the current labels.

2.3.4. Weakly Supervised Learning

Weakly supervised learning is aimed to take the strengths of supervised and unsupervised learning and train a better model. While training is provided with some labeled examples as in supervised learning, then the model tag unlabeled instances as in unsupervised learning. Unlike supervised learning, weak learning tries to give the model a small number of labeled data from the beginning and try to get the model to extract more examples from the text. Weak supervision aims to use cheaper sources of labels that can be noise or heuristic. Unlike semi-supervised learning, it tags unlabeled data using its own knowledge. In other words, the model discovers new knowledge while doing the labeling process.

Weak supervision is subdivided into crowdsourcing, distant supervision, heuristic and other approaches. The most popular of these is the distant supervision approach [32]. Distributional module and pattern module of our model use distant supervision approaches.

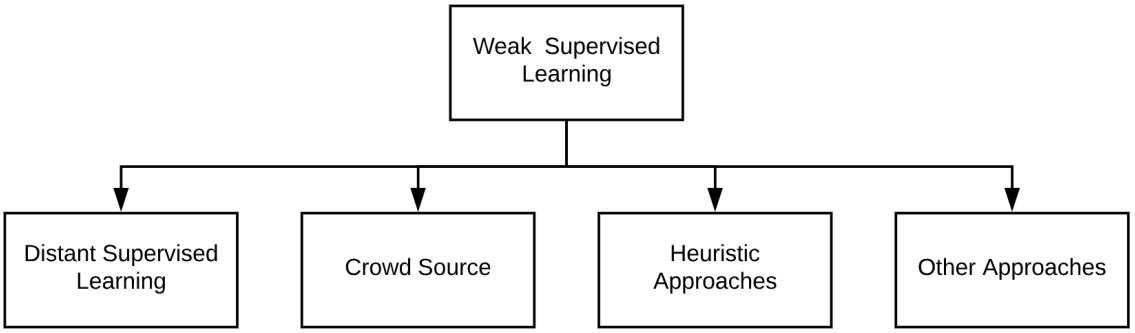

Figure 2.3 Classification of Weak Supervised Relation Extraction

2.3.4.1. Crowdsourcing Unlabeled sentences can be tagged for relation extraction using people. Crowdsourcing is a cost-saving method used to create labeled datasets in machine learning. Amazon Mechanical Turk can be given as an example for crowdsourcing. Users choose a task and these tasks are usually easy and independent from each other, such as tagging a picture, marking if there is a dog in the picture. The person receives a small fee for the tasks done. Not all users are equally reliable. While some spamming, others may knowingly give wrong answers. For this reason, even though they are tagged by humans, crowdsourcing datasets may have incorrect or missing tags.

2.3.4.2. Distant Supervision Distant supervision is predicated on the assumption that if two named entities exhibit a particular relation in the knowledge base, then any other sentence in which the same name entities feature is perhaps indicative of the same relationship. Knowledge bases, such as Wikipedia and Freebase, are utilized by distant supervision. Distant supervision models are trained through the utilization of knowledge bases.

2.3.4.3. Heuristic Approaches In this approach, relations are tried to be extracted from sentences using the predefined patterns and rules created by SME.

2.4. Data Programming

The weak supervision approach involves the amalgamation of data sets in order to train the model. However, there exist two primary challenges associated with this technique. Firstly, the data procured from diverse sources may be contradictory in nature. Secondly, it is imperative to transfer crucial lineage information pertaining to label quality to the model. This novel approach is utilized for the automatic labeling of data. Specifically, the technique of data programming is employed for this purpose. It extracts relations using the predefined rules created by SME and knowledge from knowledge base. Simple functions written in large numbers are applied to each instance of the dataset. These functions are called labeling functions [33]. These functions either give the sample a label or abstain. Different functions may give different labels to the same instance, which contradicts each other. Finally, the results of all functions are summed up and the label that the majority says is accepted as the label of the sample. Unlike heuristic approaches, in data programming, decisions are not made only by looking at the predefined rules, but also information from knowledge base is used. Thanks to data programming, the time and cost of data labeling can be greatly reduced. In addition, instead of tagging the training dataset with the information of the SMEs, the field information in data programming is kept in the functions in a reusable and updateable way [5].

2.5. Dependency Parsing

The process of extracting the syntactic structure of a sentence and displaying the interrelationships among the words in the sentence is commonly employed. Words are divided into two categories, heads and dependencies, and their relations with each other are shown with links. Head words are the crucial part of the sentence. Dependency words provide clarifications or complements to head words [34]. By using the links between the

selected words, a structure called dependency tree. Dependency tree shows the links in the sentence [35]. The direction of the arrow shows the relation between words. The link comes from head to dependent. Dependency tree, which is created according to the links of the words with each other, shows the structure of the sentence. A label is put on the link to show the type of relation. Dependency parsing is frequently used for grammatical and syntactic analysis, but these analyzes do not provide an in-depth and rich view of sentence structure.The depiction of the sentence's dependency tree is illustrated in Figure 3, which has been generated through the utilization of CoreNLP[1]. The verb is the root of the sentence and other words are directly or indirectly related to it. The utilization of Dependency Parsing is integral to a unit of the proposed model. In this model, the dependency tree of the sentence is ascertained, and the positions of the named entities on said tree are determined. Subsequently, the shortest path between the identified entities is detected. The retrieval of this path is referred to as the Shortest Dependency Path. The lexical bundle present within the Shortest Dependency Path is labeled as a Pattern. The Pattern Module of the proposed model is responsible for generating these patterns, and the syntactic data regarding the relationships is endeavored to be extracted from them.

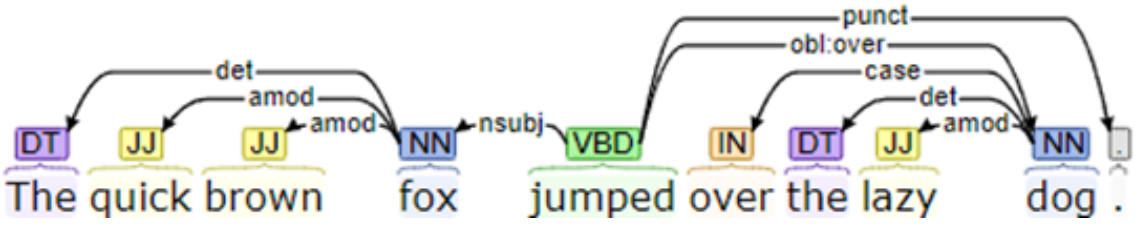

Figure 2.4 Dependency Tree Example

2.6. Text Vectorization

Texts must be converted into a mathematical input so AI models can work on it. Thus, words turn into a structure that AI models can work on. This process is called text vectorization. The texts given as input may consist of a few words or a whole book, but the vectors must be in a standard structure. Each property of the vector is called a feature [26, 35].

[1]https://corenlp.run/

2.6.1. Word2Vec

Word2Vec[1] is an unsupervised neural network study that measures the semantic similarities of words according to their vector embeddings. It takes a huge dataset as input and produces vector space as output. Vector embeddings are created for each unique word in the input and the words are placed somewhere in vector space. It uses two different architectural models to create vector embeddings. These are Skipgram and Continuous Bag of Words (CBOW). The neural network is connected to the results obtained from these models and the softmax classifier is used as the activation function in the output layer. Word2Vec is used in the proposed model as distributional module. In the CBOW model, tries to predict target word by analyzing context words. Each word in the sentence is selected as target, respectively, and the words in front and behind are taken as inputs. Then, target word is guessed. The number of words taken as input is called window size. Since the order of the words is not important in this model, it is called bag-of-words. In Skipgram, as in CBOW, each word in the sentence is placed in the center in order. However, unlike CBOW, this time context words are tried to be guessed by looking at the target word. CBOW is trained faster and performs better with frequent words. On the other hand, Skipgram performs better for rare words and it gets semantic relations between the words. Figure 4.1 shows the structures of CBOW and Skipgram.

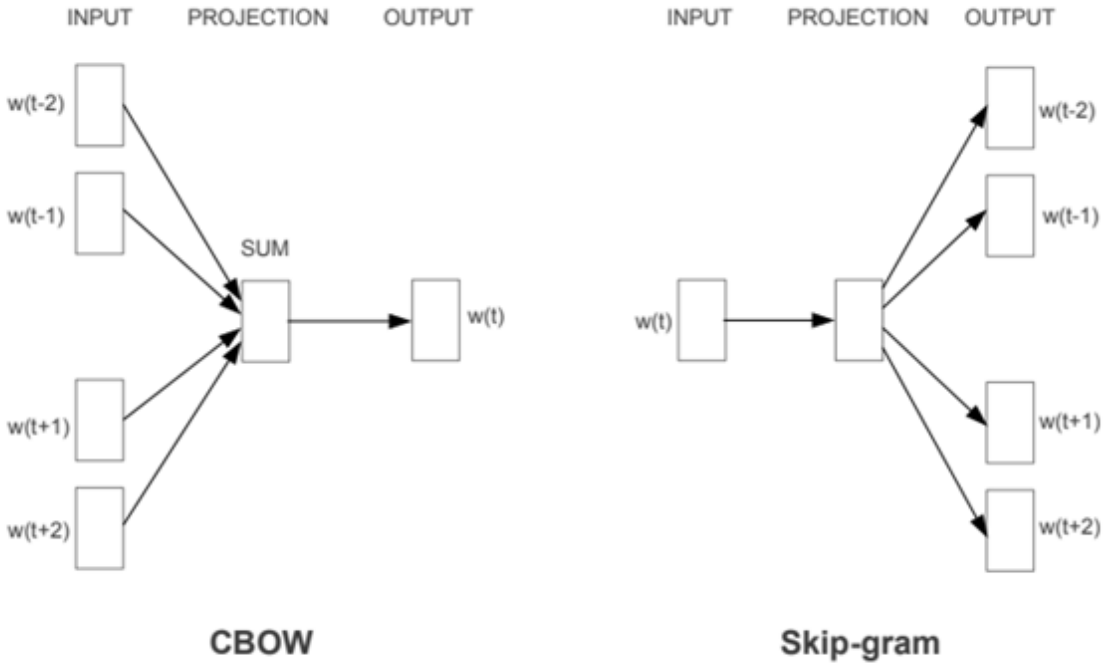

Figure 2.5 Architectures of CBOW and Skipgram[1]

3. LITERATURE REVIEW ON WEAKLY SUPERVISED RELATION EXTRACTION

Weakly supervised relation extraction is extracting relations with data that is less costly than labeled data but is also less reliable and noisy. Weak supervision is a broad concept that includes different methods as mentioned in Chapter 2. There are four subsections of weak supervised learning. These are distant supervision, crowdsourcing, heuristic approaches and other approaches. In this section, some studies on these subsections are examined.

3.1. Distant Supervision

Distant supervision is one of the sub-titles of weak supervision. Smirnova and Cudré-Mauroux [36] classified distant supervised relation extraction studies under three main headings. These are noise reduction approaches, embedding-based approaches and leveraging auxiliary information. In the noise reduction approaches, not fully labeled datasets are automatically labeled by the model. Mention level classifiers check if there is a relationship between the named entities that are provided by knowledge base. Hierarchical Topic-based models detect lexical and syntactic patterns and try to distinguish patterns that indicate relationships and entity pairs or background text patterns. Pattern correlations models deal with syntactically separating patterns that express the relationship from other patterns. It is intended to reduce false labels by removing the list of unrelated patterns. Embedding based models make use of word embeddings for relation extraction. Convolutional neural network models also use word embeddings. The last category includes additional approaches that can assist in relation extraction. Direct supervision is manual labeling. Entity identification uses various entity information such as entity types. The process of subtracting relations according to a certain rule is called Logic Formulae. Classification used in this Chapter does not provide a definitive classification for all studies. There may be studies that fall into several different groups. The classification is shown in Figure 4.3.

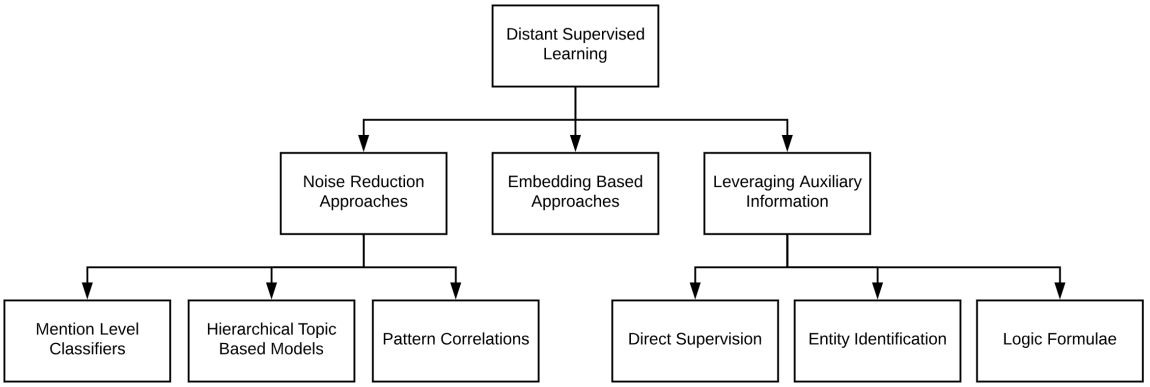

Figure 3.1 Distant Supervised Methods on Relation Extraction

Pattern module in the proposed model is a mention level classifier approach. It extracts patterns on sentence level. Distributional module is an embedding based approach because it tries to extract relations by using word embeddings. In the proposed model, we benefit from some leveraging auxiliary information approaches such as entity identification. We use NEL as entity identification.

3.1.1. Noise Reduction Approaches

Incorrect labeling may occur due to too strong distant supervision assumption or incomplete knowledge base. The error rate is tried to be reduced with noise reduction methods. Mention level and pattern learning methods deal with resolving mislabeling, while hierarchical topic based models deal with the problem of knowledge base incompleteness.

3.1.1.1. Mention Level Classifiers In the Relation Extraction basic assumption, it is said that each sentence that contains the named entities that are related to each other represents a relationship. However, sentences in which two named entities are related may not always indicate that relationship. Because this proposition is so powerful, Riedel et al. [37] used at-least-one assumption in their study. That is, at least one of the sentences containing each NE pair that has a relationship between them can indicate the relationship. In this case, it is necessary to distinguish which sentence indicates the relationship. With

the undirected graphical model they created, the sentences containing a relationship are estimated. Afterward, constraint-driven semi-supervision model was applied. 1000 Freebase instances were extracted from the NYT dataset. At-least-one assumption reached 91% accuracy and 87% precision. This means that the erroneous prediction rate has decreased by 31% with expressed-at-least-one assumption.

Hoffmann et al. [38] introduced a novel methodology for multi-instance learning that incorporates overlapping relationships. Their perspective on distant supervision relation extraction as a multi-instance learning quandary that denotes relationships based on multiple sentences, rather than a solitary sentence, was groundbreaking. They conducted experiments to ascertain the efficacy of their model in learning NY Times text extractors through weak supervision from Freebase. As a result, the approach is remarkably expeditious and leads to remarkable improvements in precision at both the aggregate and sentence levels.

Agichtein and Gravano [39] suggested the weak supervised Snowball model in their study. Patterns and tuples are extracted automatically without human intervention in the model, in which the most reliable patterns are pre-served in each iteration. They created high-quality tables with the new method they developed in their experiments using more than 300 thousand news-paper articles, and thus they increased the effectiveness of relation extraction.

Surdeanu et al. [40] aimed to solve the multi-instance multi-label learning problem with the graphical model they created. The model is tried on two challenging domains and performs well on both.

In conducting their research, Min and Wang[41] created a weakly supervised algorithm that was solely trained on affirmative and indeterminate samples. Furthermore, they demonstrated that a noteworthy proportion of the samples categorized as negative by the model were actually false negatives, a consequence of an insufficient knowledge base.

In their research endeavor, Riedel et al. [42] endeavored to produce a comprehensive schema for performing relation extraction in a universally applicable manner. The proposed framework, based on matrix factorization, enables the learning of latent feature vectors for

both entity tuples and relationships. Evaluation of the approach against the State-of-the-art Distant supervision model via the Mean Average Precision metric demonstrated an improvement of 10% points in efficacy.

Fan et al. [43] considered weakly supervised relational extraction as an incomplete multiclass classification problem characterized by sparse and noisy features and proposed the low-rank matrix completion technique as a remedy. In contrast, they offered substantial enhancements in performance relative to alternative weakly supervised algorithms.

3.1.1.2. Hierarchical Topic Based Models In texts, the word topic is generally used to express words or patterns that occur together. The purpose of topic models is to include each term in a topic. Mention level classifiers evaluate each sentence individually, while topic-based models can detect a more general relationship between relational and textual patterns [36].

In their study, Yao et al. [44] clustered entities and the patterns between them by proposing series of generative probabilistic models. Expressions in each model cluster are set to give the same relationship. In the tests performed, it was seen that it made 12% less errors for precision.

Alfonseca et al. [45] performed relation extraction without manual interference using the topic model. Mode distinguishes between relationship-specific patterns and more general patterns. As a result of the experiments, the model was able to detect some relationships that are not in Freebase. It can also be used to find supporting sentences for known relationships.

Wang et al. [46] proposed a topic-based model. In the three-stage study, firstly, a relation repository with more than 7000 relations taken from Wikipedia was created. In the second step, the relationships in the repository are divided into topics, without duplication. In the third step, new relationships are detected using these topics and SVM. Tests with Wikipedia data and the ACE dataset showed that the model gave good results in relation extraction.

In their study, Roth and Klakow [47] combined the at-least-one approach, which is a noise reduction method, and the hierarchical topic-based model method. In the experiments, it was seen that combining the methods increased the relation extraction quality of the model.

Han et al. [48], have devised a novel hierarchical attention scheme by amalgamating hierarchical information of relations in the context of distantly supervised relationship extraction. Their experimental investigations on sizable benchmark datasets have led them to conclude that their approach surpasses other baselines significantly.

3.1.1.3. Pattern Correlations Pattern correlation techniques aim to quantify the degree of correlation between a given pattern or set of patterns and a particular relation or association among entities present within a sentence or text.

Mintz et al. [49] assume that if there is a relationship defined in the Knowledge Base between the entities, there is the same relationship in other sentences where the same entities are mentioned. They proposed a model and in the tests performed, 10000 instances were extracted from 102 relations with 67.6% accuracy.

Takamatsu et al. [50] tried to reduce the number of incorrectly labeled data in their study. They proposed a generative model that tells whether the pattern contains a relationship by looking at hidden variables. In the tests performed, it was seen that the model detected the faulty tags better and reducing the erroneous tags increased the relation extraction performance.

Kirschnick et al. [51] proposed a resource called Freepal, designed to help relation extractors. The aforementioned model is comprised of an excess of 10 million lexico-syntactic patterns. Furthermore, the CLUEWEB09 dataset has been utilized to annotate in excess of 260 million sentences containing entities and relationships in Freebase.

3.1.2. Embedding Based Approaches

Embedding-based methodologies endeavor to derive meaning from textual representations of entities and relationships by embedding them in word space. However, embedding vectors alone may not suffice to preserve intricate word relationships and properties. To address this issue, Xu et al. [52] proposed a novel framework known as RC-NET, which entails a distinct regularization function applied to relational and categorical knowledge, and subsequent integration of the results through the objective function in Skipgram. The approach leverages backpropagation neural network to optimize the problem at hand, while also augmenting word representations via the incorporation of categorical knowledge. Experimental evaluations conducted in prominent data mining and natural language processing domains, including but not limited to analogical reasoning, word similarity, and topic prediction, have demonstrated that RC-NET significantly improves the overall quality of word representations.

Raj et al. [53] proposed a two-layer model called a weak supervised convolutional recurrent neural network in order to extract relations in biomedical texts. These layers consist of CNN and RNN. CNN was used for coarse-grained local feature detection in the sentence, while RNN was used to extract long-term dependencies. The outcome of their empirical investigations evinces that the posited framework attains preeminent efficacy in respect of two distinct datasets.

According to Zeng et al. [54] In their study, they use a neural model to extract relationships using inference chains that use intermediate entities to reveal the relationship between target entities even if they could not pass the same sentence. Through conducting experiments on actual datasets, it was discovered that the aforementioned model is capable of effectively utilizing sentences that exclusively feature a singular target entity. Furthermore, it was observed that this model yields marked and dependable enhancements in the domain of relation extraction, in comparison to established benchmarks.

Lin et al. [55] created a model to solve the wrong labeling problem encountered in distant supervised relation extractions in their experimental study. First, CNNs are used to embed the semantics of the sentences. Sentence-level attention was then applied to dramatically reduce the weight of the noisy instances. As a result of their experiments, they observed that the model they developed significantly and continuously improved the performance of correlation extraction compared to baselines.

Vashishth et al. [56] proposed a weakly supervised method for extracting neural relationships called RESIDE. They used Graph Convolution Networks in their study. Each word in the sentence is accepted as a node in the graph and a dependency tree is used to create structures between the nodes. Through extensive experiments with benchmark datasets, they demonstrated the effectiveness of RESIDE.

Lin et al. [57] proposed an attention-based, sentence level relation extraction model to alleviate the mislabeling problem. First, the semantics of sentences are extracted with a convolutional neural network. Then, the effect of noisy labels was tried to be alleviated with sentence-level attention. In the tests performed, it was seen that it gave better results than the baseline.

Relation extraction typically relies on statistical approaches, thus its effectiveness is contingent upon the features that are extracted. These features are commonly derived from the outputs of established natural language processing (NLP) systems. Bugs are also reflected in the tools used and the result of the model. Zeng et al. [58] aimed to extract features at the lexical and sentence level by using the convolutional deep neural network in the proposed model. The softmax classifier checks whether there is a relationship in the sentence with the features obtained from these two levels. The results showed that the model had significant improvements.

Lin et al. [59] developed the DualRE model in their study to improve the performance for weakly supervised relation extraction. The DualRE model consists of prediction and retrieval modules. These modules make RE by mutually improving each other. While the neural relation extraction model used in the prediction module aims to extract a relationship

from the sentence, the retrieval module tries to find example sentences for the relationship with the neural learning-to-rank model. It has been shown by their tests that they increase the RE effectiveness of the proposed model.

In their study, Jiang et al. [60] proposed a multi-instance multi-label convolutional neural network model utilizing weak supervision. Through the utilization of the cross-sentence max pooling method, the model enables the sharing of information across sentences for relation extraction. According to their experimental results, their approach consistently and significantly surpasses current state-of-the-art methods.

3.1.3. Leveraging Auxiliary Information for Supervision

Various distant and direct supervision techniques are employed to enhance the efficacy of relation extraction procedures through the integration of supplementary information or expertise.

3.1.3.1. Manual Labeling In this method, it was tried to increase the success of weak supervision by adding hand-labeled samples to the training data.

Angeli et al. [61] combined some hand-labeled samples with a large corpus of distantly labeled data. In their experiments, they observed that the F1 score of the models improved by 3%.

In their study, Pershina et al. [62] proposed a model. In addition to the relation level distance supervised database to the multi-instance multi-label model, sentence level labeled examples were given. In the conducted evaluations, a noteworthy enhancement of 13.5% was attained in the F-score metric, whereas a substantial upsurge of 37% was observed in the area covered by the precision-recall curve.

3.1.3.2. Entity Identification Entities in sentences are extracted via NER. Thus, canonical entities are found. However, the same assets may be referred to by a different name

or abbreviation. Names that describe the same entity with the NEL process can be mapped to a single entity. Entity names are capable of being categorized as noun, pronoun, or noun phrases. In certain instances, it may become imperative to reference an entity mentioned in the preceding sentence, despite the absence of direct mention. Also, knowing the types of entities can be helpful for relationship inference.

In their research, Augenstein et al. [63] employed distantly supervised relationship extraction techniques on Web-based data. However, owing to data sparsity, noise, and lexical ambiguity, they could not achieve satisfactory performance. To address these issues, they utilized a more robust tool such as Named Entity Recognition (NER) and reduced data sparsity. As a result, an 8% increase in precision was achieved. In addition, strategic training data was selected using statistical methods to reduce noise, and an additional 3% increase in precision was achieved.

Koch et al. [64] examined the increase in the performance of the weakly supervised model with NEL and coreference added. In tests with 48 relation classes in NYT and GORECE datasets, it was observed that precision increased by 44% and recall by 70%.

Tonon et al. [65] tried to predict the rank of the given asset according to the context. They tried to determine the type of entity with statistical methods on a graph that connects entities and their types. A regression model trained with this approach reached the Mean Average Precision value of 0.70.

In their study, Milne and Witten [66] add cross references to Wikipedia pages automatically. The link detector and disambiguator have reached almost 70% precision and recall. It has been stated that the proposed technique can be used for tasks solved with bag of words.

3.1.3.3. Logic Formulae In this method, extractions are made according to the connections between the relations. For example, entities with a “capitalOf” relationship cannot have a “childOf” relationship.

Although matrix factorization is a successful method for relation extraction, if there is no sample related to the target relation in knowledge-based, it cannot perform relation extraction. Sparse data is a problem. On the other hand, rule-based extractors can capture new relations with first-order formulae. However, these formulae need to be generalized in large data sets. Rocktaschel et al. [67] combined matrix factorization and first-order formulae methods. Experiments show that model can make learn extractors with little or no distant supervision. Moreover, the model can generalize textual patterns that do not appear on formulae.

Distant supervision is typically beset by inadequate supervision. In their study, Han and Sun [68] posited a comprehensive model for distant supervision which alleviates the paucity of supervision by furnishing an array of indirect supervisory information.The model also reduces uncertainty in distant supervision. In the conducted experiments, the model significantly outperforms traditional distant supervision approaches on a publicly available KBP data set.

3.2. Crowdsourcing

Thanks to crowdsourcing, collective knowledge and different backgrounds of individuals can be used for data labeling. Although this method reduces the cost of labeling data, it is expected that the quality of the labeled data will also decrease.

Yuen et al. surveyed the different usage areas of crowdsourcing in their study. These tasks include NE annotations and NLP annotation [69].

3.3. Heuristic Approaches

In this method, relation extraction is performed according to domain specific rules, patterns or heuristics.

Rekatsinas et al. [70] introduced the HoloClean framework in their study. Holoclean is heuristic data repairing framework. It combines existing qualitative data repairing

approaches with quantitative data repairing methods. When inconsistent input is given, HoloClean creates a probabilistic program that automatically repairs data. In the experiments, HoloClean repaired data with an average of 90% precision and 76% recall values in data sets containing millions of records and different errors. HoloClean provides approximately twice the F1 improvement compared to current methods.

3.4. Other Approaches

Uncategorized studies are included under this heading.

Zaidan and Eisner [71] propose a new generative method. With this method, they showed how they could extract extra information in the form of rationales from naive annotators. In tests, the method outperformed the two strong baseline classifiers.

In their analysis, Mann and McCallum [72] investigated generalized expectation criteria, which is a technique for semi-supervised learning trained on weakly labeled data. The authors demonstrated the applicability of generalized expectation criteria to maximum entropy models and conditional random field models. Empirical findings have indicated that this approach outperforms contemporary semi-supervised learning methods for these models. Conditional random fields trained with label regularization improved by 1% to 8% compared to purely supervised approaches.

4. PROPOSED WEAK SUPERVISED RELATION EXTRACTION MODEL

In this Chapter, REPEL relation extraction algorithm, which is the source of inspiration for the study, and our proposed model are explained.

4.1. REPEL

The REPEL algorithm, a moderately effective approach to supervised relation extraction, comprises two distinct modules: a pattern module and a distributional module. These modules provide varying perspectives on sentence analysis, affording the algorithm the capacity to extract relations from differing angles. While the pattern module evaluates the sentence in terms of syntactic perspective, the distributional module analyzes semantically. Because REPEL is a weak supervision algorithm, it needs a small group of samples at the beginning to train the modules. However, unlike other weak supervision models, the two modules provide extra supervision by giving feedback to each other and providing extra supervision for their training. This strategy is called co-training [73]. The Pattern module serves as a generator by extracting potential relation instances from sentences. Distributional module can be called a discriminator because it evaluates the instances found by pattern module. These evaluations from the discriminator provide feedback for the generator. On the contrary, the generator plays a vital role in facilitating the instruction of the discriminator by detecting and presenting more dependable relation instances. REPEL works in cycles, and continues to train itself with the modules feeding each other at the end of each cycle. Since REPEL is a weak learning algorithm, labeled data is required to work on the model. For the relations to be extracted, a seed document with examples of NE pairs is required. The general purpose of REPEL is shown in equation (1).

$$\max \mathrm{O}_{P,D} = \max \mathrm{O}_{P,D} \left\{ \mathrm{O}_p + \mathrm{O}_d + \lambda \mathrm{O}_i \right\} \tag{1}$$

In equation (1), the symbol P denotes the parameters of the pattern module, specifically representing the number of reliable samples for each target relation. Symbol D, on the other hand, refers to the entity representation and score function, which are the parameters of the distributional module. The objective of the pattern module, as indicated by symbol O_p, is to select a reliable pattern from the given seeds. Meanwhile, the distributional module aims to select the appropriate parameters based on the provided seeds, as denoted by symbol O_d. Finally, the interaction of the two modules is represented by symbol O_i.

4.1.1. Pattern Module

The objective of the pattern module is to identify a predetermined quantity of dependable patterns for the intended relation and subsequently extract additional instances of relationships by utilizing the aforementioned patterns. Within the pattern module, the sentence is assessed syntactically. The shortest dependency path between designated entities is parsed, which is referred to as a pattern. Patterns are extracted from all sentences which have at least two named entities. It checked how many NE pairs from the same pattern are in the seed document. The more reliable examples the pattern has in the seed document, the higher the score it gets. For example, suppose two instances with a candidate pattern are discovered in the text for which the relation “capitalOf”. Let these instances be (“berlin”,”germany”) and (“ankara”,” turkey”). (“ankara”,”turkey”) is in the seed document, whereas (“berlin”,”germany”) is not. In this case, the score for the candidate pattern is 1/2. In Figure 4.1 how pattern module works can be seen.

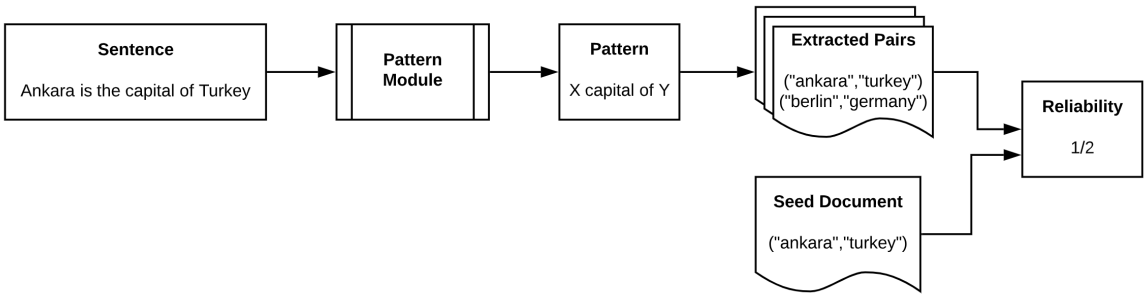

Figure 4.1 Overview of Pattern Module

The reliability score function of the Pattern module is presented as equation (2). Herein, π symbolizes the pattern, and R represents the reliability score of said pattern. G refers to the set of extracted instances of the pattern from dataset. S_{pair} refers to the cluster of seed instances.

$$R(\pi) = \left| G(\pi) \bigcap \mathrm{S}_{pair} \right| / G(\pi) \tag{2}$$

Let us consider K to be the quantity of patterns to be extracted as a parameter of the pattern module. In this scenario, the objective function of the pattern module adheres to the specifications outlined in equation (3). P denotes the collection signifying the quantity of K patterns.

$$\mathrm{O}_p = \sum_{\pi \in P} R(\pi) \tag{3}$$

The utilization of K most reliable patterns presents a viable method for the identification of novel entity pairs subsequent to their extraction. This situation is shown formally in equation (4). G(P) represents the extracted NE pair of patterns and P is set of patterns.

$$G(P) = \mathrm{G}(\pi)_{\pi \in P} \tag{4}$$

4.1.2. Distributional Module

The Distributional module learns the representations of entities from the text and assumes that entities in a similar state will have similar meanings. NE pairs are analyzed semantically in distributional module. In REPEL distributional module uses a bipartite network. It is established between all words and entities [74]. The weights of the connections between entities and words are determined according to the number of sentences they belong together. Then the conditional probability between entity and word is determined as in equation (5).

$$P\left(w|e\right) = exp\left(\mathrm{x}_e \cdot \mathrm{c}_w\right)/Z \tag{5}$$

In accordance with equation (5), the variable w denotes a word, while e denotes an entity. The representation vector of an entity is symbolized by x_e, whereas c_w signifies the word embedding for a given word. Z is used for normalization. The obtained objective function for text is shown in equation (6). $\mathrm{n}_{w,e}$ refers to the edge between the word and the entity.

$$\mathrm{O}_{text} = \sum_{w,e} \mathrm{n}_{w,e} logP\left(w|e\right) \tag{6}$$

In addition to the text, NE pairs in the seed document are used for the score function. Using the work of Bordes et al. [75], the score function for the target relationship for an entity pair is stated in equation (7). e_h and e_t are the representation vectors of entities, while r is the parameter vector showing the target relationship. $\|\|_2$ is the Euclidean norm of vector. f represents NE pair of h and t. f= $\left(\mathrm{X}_{\mathrm{e}_h}, \mathrm{X}_{\mathrm{e}_t}\right)$

$$\mathrm{L}_D\left(f|r\right) = \left\|\mathrm{X}_{\mathrm{e_h}} + \mathrm{rX}_{\mathrm{e_t}}\right\|_2^2 \tag{7}$$

Seed entity pairs are naturally expected to score higher than random entity pairs. For this reason, ranking-based objection function given in equation (8) is used for training.

$$\mathrm{O}_{seed} = \sum_{f \in \mathrm{S}_{pair}} \sum_{f=\left(\mathrm{e}_h, \mathrm{e}_t\right)} min\left\{1, \mathrm{L}_D\left(f|r\right)\right\} \tag{8}$$

S_{pair} refers to the set of all entity pairs in the seed. e_h and e_t refers to the set of random entity pairs. The sum of the objection functions in equation (6) and (8) creates the objection function of the distributional module in equation (9).

$$\mathrm{O}_d = \mathrm{O}_{text} + \mathrm{O}_{seed} \tag{9}$$

Since the L_D function can be used to measure the scores of entity pairs under the target relationship, it can be used to discover entity pairs with high reliability.

With the help of word embeddings, it is calculated how entities are similar to each other, and all patterns are scored according to the closeness of named entities. Scores from distributional module and pattern module are added at the end of each cycle. Top K pattern with the highest score is considered reliable. The distributional module is retrained with examples of the selected reliable patterns and if any of these NE pairs are not in the seed document, they are added to the document. Then a new cycle is started.

4.1.3. Modeling the Module Interaction

The interaction of pattern and distributional module is using co-training method. equation (10) is given as an objective function of O_i. The G(P) is a set of relation instances extracted by the module. $\mathrm{L}_D\left(f|r\right)$ is the score function of the f pair instance under the target relationship.

$$\mathrm{O}_i = \mathrm{E}_{f \in G(P))}\left[\mathrm{L}_D\left(f|r\right)\right] \tag{10}$$

The objective function serves the purpose of merging the pattern and distribution modules in the most optimal manner conceivable. To achieve this, the pattern module must opt for instances that the distribution module trusts. This implies that the instances with higher scores in L_D, the distributional module's score function, ought to be chosen to maximize the function in equation (10). On the Distributional module's side, it is paramount to assign a high score to the instance pairs that the pattern module has selected. This is because the pattern module's highly reliable instance pairs provide the necessary support for the training of the distributional module. By employing the objective function in equation (10), both modules offer additional supervision to each other and attempt to mitigate the adverse effects of insufficient labeled data.

4.1.4. The Joint Optimization Problem

REPEL model works as iterative. The coordinate gradient descent algorithm is used to optimize the function in equation (1) [76]. The algorithm, which works in loops, works as two subprocesses. The optimization steps of REPEL are given in Figure 4.2.

Algorithm 1 REPEL Algorithm

Input: A text corpus, a few seed relation instances, the number of reliable patterns K, the parameter λ.

Output: A set of reliable patterns P from pattern module, a score function L_D from distributional module, extracted relation instances.

Generate patterns and entity pairs extracted by each pattern.

Build the bipartite network between entities and words.

while *not converge* **do**

  Update the distributional module:

  Extract some instances by using the set of reliable patterns P.

  Optimize D with both the seeds and extracted instances (equation (11)).

  Update the pattern module:

  Calculate pattern reliability with the seeds and L_D (equation (12)).

  Select the top-K most reliable patterns to form the pattern set P.

end

Extract relation instances:

Utilize the reliable patterns P to extract instances from local contexts.

Utilize the distributional score function L_D to extract instances

Figure 4.2 REPEL Algorithm[2]

In the first subprocess of Figure 4.2, the pattern module does not change and the distributional module is updated with the instances in the seed and the confident instances found by the pattern module. The optimization function used in equation (11) is shown. Scholastic gradient was used for continuous optimization problem. The first part represents objective function of distributional module and the second part represents objective function of interaction. NE pair instances of reliable patterns act as extra seeds for the update.

$$\max_D \{\mathrm{O}_d + \lambda \mathrm{O}_i\} = \max_D \left\{ \mathrm{O}_d + \lambda \mathrm{E}_{f \in G(P))} \left[\mathrm{L}_D \left(f|r \right) \right] \right\} \tag{11}$$

In the second subprocess of Figure 4.2, the distributional module remains fixed and the pattern module updates the seed instances and the patterns selected with the extra supervision provided by the distributional module. This creates the optimization problem shown in equation (12). This is a discrete optimization problem. R(π) is as described in equation (2). O_i part is to put all instances in G(π) into distribution module score function.

$$\max_P \{\mathrm{O}_d + \lambda \mathrm{O}_i\} = \max_P \left\{ \sum_{\pi \in P} R\left(\pi\right) + \lambda \mathrm{E}_{f \in G(P))} \left[\mathrm{L}_D \left(f|r \right) \right] \right\} \tag{12}$$

4.2. Proposed Weak Supervised Relation Extraction Model

The proposed model consists of three main components which are pattern module, distributional module, and labeling functions module as can be seen in Figure 4.3. Just like in REPEL, the pattern module analyzes the sentence syntactically, while the distributional module looks at the semantic similarity between named entities. On the other hand, the newly added labeling functions module examines the sentence from a lexical perspective and brings a new perspective to the relation extraction process. The Pattern module acts as a generator and produces the most reliable K pattern. However, this time there is both a distributional module and a labeling functions module on the discriminator side. Unlike REPEL, the generator not only transmits the NE pairs it extracted but also the patterns of these instances to the discriminator side. Distributional module interest with NE pairs and labeling function modules analyzes both NE pairs and pattern itself. The objective function of the algorithm is shown in equation (13). A new part is added to the objective function of REPEL. This new part shows the interaction of pattern module with labeling functions module.

$$\max \mathrm{O}_{P,D,L} = \max \mathrm{O}_{P,D,L} \left\{ \mathrm{O}_p + \mathrm{O}_d + \lambda \mathrm{O}_i + \mu \mathrm{O}_j \right\} \tag{13}$$

There is a seed document that includes some ground-truth relation instance examples because the proposed model has weak supervised modules which are pattern and distributional modules. Additionally, labeling function modules need some input value to execute rule-based functions. Weak supervision models need seeds because they need initial information to tag unlabeled instances. In Figure 4.3, database represents the seed document. There are NE pairs and some words that related to the relation name in the seed document. NE pairs are used by pattern and distributional modules and words are used by labeling functions module. In addition to seed instances, label words can be used as evaluation criteria. Label words are specific words that strongly evoke a relationship.

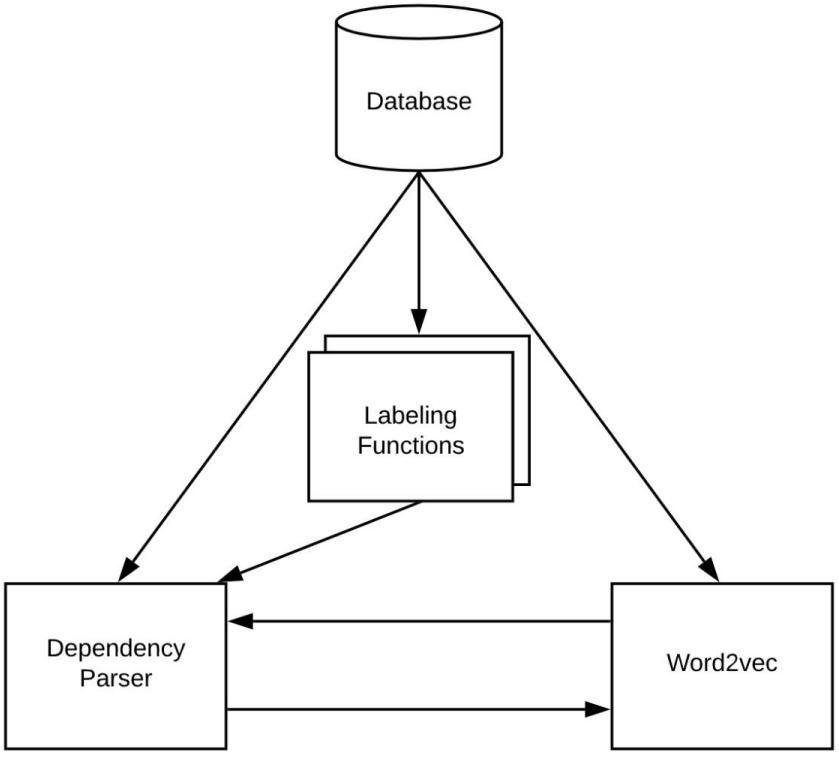

Figure 4.3 Architecture of the Proposed Model

4.2.1. Pattern Module

Pattern module generates candidate patterns, and the other parts examine whether this pattern is meaningful. The highest scored candidate patterns are accepted as reliable patterns. Although the pattern module generally works the same as in REPEL, there are some important differences. Pattern module in the proposed model has an enhanced reliable score function. The objection function is generally the same as in equation (3). The newly learned instances are also extracted as in equation (4). Pattern module sent not only instances but also patterns to the discriminator side. Instances are kept as mapped to patterns. Thus, it can be known which instances are extracted with which pattern.

RE pipeline starts with sentence segmentation. The raw text was segmented into sentences by using regular expressions. After that, tokenization process is applied to each sentence. Tokenized sentences are tagged with part of speech (POS) labels. By using POS tags, named entities in the sentence are detected. Tokenization, POS tagging and NER are made by using SpaCy [77] language model tool in English model. There is no offical SpaCy model for Turkish. Because of that another language model is used[2]. Some of the named entities may represent the same entity. To make the model run more efficiently, all named entities should be mapped to unique entities. Dbpedia tool[3] is used for NEL. After preprocessing, sentences with fewer than two named entities are excluded from the input because there is no possibility of finding a relation. The proposed model extract relations on sentence level.

Pattern module is a dependency parser that determines the shortest path between the selected NE pair. All the candidate patterns are detected at the beginning. NE pairs in the structure of the candidate pattern are extracted from the text. Pattern module checks how many of the extracted pairs are in the seed document and gives a score to the candidate pattern with respect to the number of common pairs. In each iteration, pattern module selects the highest-scored candidates according to equation (14). And sends selected candidates to other

[2]https://huggingface.co/spaces/akdeniz27/spacy-turkish-demo
[3]https://www.dbpedia.org/

modules. Pattern module can send single or multiple candidates according to the parameter adjusted at the beginning.

$$R(\pi) = \left(\left| G(\pi) \bigcap \mathrm{S}_{pair} \right| / G(\pi) \right) + y\left(G(\pi)\right) \tag{14}$$

In equation (14), π refers to the candidate pattern. R is the reliability score of the candidate. G refers to the cluster of found instances of the candidate pattern in the dataset. S refers to the cluster of seed instances. Score function used in REPEL gives high scores to the candidate patterns when the candidate pattern has few instances and these instances are in the seed document. This prevents the discovery of new instances. Another problem is that this score function accepts every pattern with the same NE pair as meaningful. However, not all patterns with the same NE pair may be significant. RE basic assumption is too strong. We use at-least-one assumption instead of it. For instance, "Ankara is another city of Turkey just like Istanbul and Izmir." Sentence has ("ankara","turkey") NE pair. Although ("ankara","turkey") in the sentence does not contain "capitalOf" relationship information, it is interpreted in this way.

Unlike REPEL, an extra part has been added to the score function in the proposed model. To promote finding new instances different from seed instances, a penalty method is added. Penalty method encourages pattern module to find new and unseen seeds. Penalty method is selected as common logarithm after conducted experiments which is mentioned in Chapter 4.

4.2.2. Distributional Module

Distributional module analyzes the similarity of the entities. It checks the similarity of the NE pair to each other in every single instance of the candidate patterns. If those named entities have a relationship with each other, they should have similar word embeddings. If a

pattern has more than one NE pair, it is averaged as a score. Equation (15) shows the score function of distributional modal.

$$\mathrm{L}_D\left(f\right) = S\left(\mathrm{x}_e, \mathrm{x}_t\right) \tag{15}$$

In equation (15), f refers to NE pair x_e and x_t and S function refers to the similarity score of NE pair. As can be seen in equation (15), a different score function is used than REPEL. This is because the algorithm used in the distributional module has been changed. A bipartite network is used for the distributional module in REPEL. Bipartite network is trained from the ground up with instances in the seed document and input dataset. This makes the distributional module dependent on the studied text and given examples. Since the distributional module aims to keep the global distribution of entities, instead of running an algorithm from scratch, using a well-trained model with more resources will increase the performance of the model in terms of relation extraction. In the proposed model, a pre-trained Word2Vec [1] model was used instead. Since more resources are used in the training process of Word2Vec model, it is expected that the learning ability of the model will be higher than a model that will be trained from scratch with the examples in the seed document. With the use of the pre-trained model, the learning and feedback capabilities of the distributional module were expected to be higher. Just like in REPEL, the pre-trained model continues to be trained with newly extracted samples at the end of each iteration. Differing from REPEL, In equation (15), Word2Vec does not use target relation information because it produces embedding vectors for global use.

4.2.3. Labeling Functions Module

Labeling functions are simple rules used to analyze the features of the sentence of the candidate pattern. Labeling functions can be used to increase confidence in the candidate pattern. Each function examines candidate patterns from a different direction. There are two types of labeling functions in the proposed model. First type looks for specific words in the

candidate pattern. When the labeling function finds a relation-specific words contained in the seed document, gives an extra score for this relation. For instance, sentences include “capital” words generally related to “capitalOf” relation. If there is “capital” in the candidate pattern, labeling function give a score to this candidate for “capitalOf” relation. This score is added to the total score of the candidate pattern. Distributional and pattern modules deal with named entities in the candidate pattern. On the other hand, this labeling function deals with other words in the candidate pattern. In this way, all possible resources to extract the relation can be used efficiently.

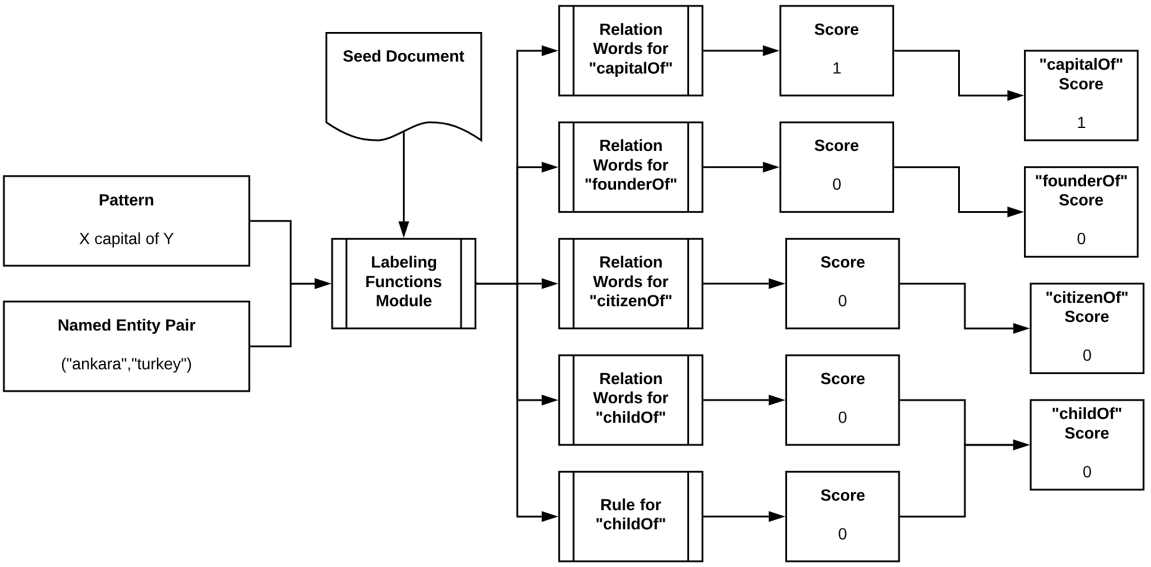

Figure 4.4 Overview of Labeling Function Module

Labeling functions do not have to be generic for all conditions. There can be functions to detect only one relationship type. In the proposed model, the second labeling function tries to detect only “childOf” relationship. It compares named entities that are “Person” and last names are the same.

Equation (16), shows the score function of labeling functions. Each function can have different score ranges. l(π,G(π)) refers to each function’s score. P refers to the set of reliable patterns and G(π) is the cluster of NE instance pairs of the extracted pattern. Since the evaluation criteria of each function will be different, score function may also differ for each labeling function. However, the rule-based structure was used for this study.

$$L\left(P, G\left(P\right)\right) = \sum_{\pi \in P} l\left(\pi, G\left(\pi\right)\right) \tag{16}$$

4.2.4. Modeling the Module Interaction

The interaction of the labeling function module is given in equation (17). μ is the coefficient of labeling functions module. Labeling functions module interacts with only pattern module. Because of that, there is no reason to change parts of objective function of distributional module.

$$\mu \mathrm{O}_j = \mu \mathrm{E}_{p \in P}\left[L\left(p, G\left(p\right)\right)\right] \tag{17}$$

4.2.5. The Joint Optimization Problem

The proposed model works in cycles. Just like in REPEL, the algorithm consists of two subprocesses. In Figure 4.5, the process is given step by step.

In the first subprocess, the pattern module does not change, and the distributional module is updated with the instances in the seed and the instances found by the pattern module on the previous iteration. The proposed model uses the same optimization function expressed in equation (11), the only difference is the used score function for distributional module. Optimization function of the proposed model is given in equation (18). The proposed model uses equation (15) instead of equation (7).

$$\max_D \left\{\mathrm{O}_d + \lambda \mathrm{O}_i\right\} = \max_D \left\{\mathrm{O}_d + \lambda \mathrm{E}_{f \in G(P))}\left[\mathrm{L}_D\left(f\right)\right]\right\} \tag{18}$$

In the second subprocess, the distributional module remains invariant while the pattern module modifies the seed instances and selected patterns in accordance with the guidance provided by the former. In contradistinction to the REPEL algorithm, this stage constitutes a

Algorithm 2 The Proposed Model Algorithm

Input: A text corpus, a few seed relation instances, the number of reliable patterns K, the parameter λ, the parameter μ.

Output: A set of reliable patterns P from pattern module, a score function L_D from distributional module, extracted relation instances.

Generate patterns and entity pairs extracted by each pattern.

Build the Word2Vec model.

while *not converge* **do**

Update the distributional module:

Extract some instances by using the set of reliable patterns P.

Optimize D with both the seeds and extracted instances (equation (18)).

Update the pattern module:

Calculate pattern reliability with the seeds, L(P,G(P)) (equation (16)) and L_D (equation (12)).

Select the top-K most reliable patterns to form the pattern set P.

end

Extract relation instances:

Utilize the reliable patterns P to extract instances from local contexts.

Utilize the distributional score function L_D to extract instances

Figure 4.5 The Proposed Model Algorithm

labeling functions augmentation, thereby conferring supplementary scores. The optimization function of the pattern module of the proposed model is represented by Equation (19).

$$\max_P \left\{ O_d + \lambda O_i + \mu O_j \right\} = \max_P \left\{ \sum_{\pi \in P} R(\pi) + \lambda E_{f \in G(P))} \left[L_D(f) \right] + \mu L\left(P, G(P)\right) \right\} \tag{19}$$

5. EVALUATION RESULTS

5.1. Datasets

In this section, information about the datasets used for model training and testing is given.

5.1.1. New York Times Relation Extraction Dataset

In New York Times Relation Extraction Dataset[4] there are 24 different relation types that each of them at least 1 instance. We used instances of “capitalOf”, “nationality”, “founderOf” and “childOf” relations in our experiments. The dataset contains 3 separate documents. These are train.json, test.json and valid.json files. Files are in JSON format. The train.json file, which has the largest number of examples, was used in the REPEL comparison experiments. The test file was used in performance analysis tests. There are 12309 examples in all of them have one of four relationships. Test.json has 1222 examples.

Due to the lack of relation extraction dataset for Turkish, New York Times dataset is used for both English and Turkish tests. Some examples from train.json file translated to Turkish. There are 2473 samples in Turkish dataset. These samples are used for REPEL comparison experiments and performance analysis tests.

5.1.2. Wikipedia

Wikipedia[5] is used to train models. Pages are shown to the model and the model extracts some patterns and instances according to the seed document. In performance tests and REPEL comparison experiments, The Turkish model is trained with 50,000, and The English model is trained with 205,328 Wikipedia pages.

[4] https://www.kaggle.com/datasets/daishinkan002/new-york-times-relation-extraction-dataset?resource=download
[5] wikipedia.org/

5.2. Evaluation Metrics

In this section, the criteria used to evaluate the tests are mentioned.

- **Confusion Matrix** The confusion matrix provides a representation of the model's performance in a classification task, and is utilized to assess the outcomes of the model's classification in the experiments.

		Prediction outcome		
		p	**n**	**total**
actual value	**p**	True Positive	False Negative	P
	n	False Positive	True Negative	N
	total	P	N	

- **Accuracy** Accuracy is used in the RE test and it is calculated how many of the classified instances are correct in total.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

- **Precision** Precision shows how many of the values estimated as positive are actually positive.

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

- **Recall** Recall shows how many of the transactions that should have been predicted as Positive were predicted as Positive.

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

- **F1 Score** The F1 Score denotes the harmonic mean of Precision and Recall measures, thereby ascertaining the accuracy of a given model.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{23}$$

- **Number of Average Guess** It shows the average number of times the model predicts until finds the correct answer. This metric was used in the KBC test. In the KBC test, capital names are shown to the model and the model responds to the semantically closest. If the answer is not the correct country name, the model returns the next closest word as the answer. The model continues to predict until the correct answer is returned.

5.3. Performance Analysis

Some tests were carried out to adjust the input parameters of the proposed model. According to the results obtained in the tests, the most optimal parameters were determined and These were set as the parameters of the proposed model in the REPEL comparison experiments. In these tests, classification was made for 4 different relations on the New York Times dataset. For English tests, 1222 samples selected from test.json were used. In the Turkish tests, 2473 samples were used. In the tests performed, the classification success of the model was measured with the F1 score. In the tests, the model is first trained with Wikipedia pages. Then, the model to classify the test data.

5.3.1. Penalty Method

As mentioned in Chapter 3, the proposed model uses a penalty method to encourage the pattern module to find new instances. Different alternatives were evaluated in order to be used as a reliable score function in the pattern module of the proposed model. For the tests, the model was run in 1 iteration and extracted 1 pattern. The coefficients of labeling function and distributional functions are selected as 1.

As can be seen in Table 5.1, logarithm and natural logarithm for English have the same F1 scores. Therefore, the common logarithm function was chosen as the penalty method for REPEL comparisons in English.

Function	**F1 Score**
log10	**0.5844**
ln	0.5844
log2	0.584

Table 5.1 Penalty Method for English

In the tests performed for Turkish, the natural logarithm received the highest F1 score. For this reason as can be seen in Table 5.2, the natural logarithm penalty method was chosen in the Turkish REPEL comparison experiments.

Function	**F1 Score**
log10	0.7646
ln	**0.7657**
log2	0.7652

Table 5.2 Penalty Method for Turkish

5.3.2. Number of Extracted Patterns

Pattern module outputs a certain number of patterns in each iteration. In the tests performed here, the effects of the system on relation extraction ability were tested by changing the number of extracted patterns. The remaining parameters were fixed in order to provide a controlled experimental environment. The proposed model ran 1 iteration. Distributional and labeling function module coefficients were given as 1. Logarithm function was selected as the penalty method for English. Natural logarithm was selected for Turkish. Relation classification F1 score was used as a comparison metric.

As shown in Table 5.3, F1 scores are very close to each other. However, according to the results, the optimal number of patterns to be obtained in one iteration is measured as 3. With

the increase in the number of patterns, it has been observed that there is a slight improvement in the model as can be seen in Table 5.3.

# Of Patterns	**F1 Score**
10	0.5838
5	0.5841
3	**0.5842**
1	0.584

Table 5.3 Number of Extracted Patterns for English

Just as evaluations in the English language, the F1 scores for Turkish are in close proximity. As a result of the limited scale of the Turkish dataset, the mean F1 scores of Table 5.4 exhibited a degree of similarity. Nevertheless, it is discernible that there is a favorable association between the number of patterns derived within a solitary iteration and the F1 score.

# Of Patterns	**F1 Score**
10	**0.7656**
5	0.7652
3	0.7649
1	0.7646

Table 5.4 Number of Extracted Patterns for Turkish

5.3.3. Number of Iterations

The proposed model works in iterations. In the tests, the effect of the model on relation extraction ability was investigated by changing the number of loops. Tests were made with 1 pattern extraction. The common Logarithm function was selected as the penalty method for English tests and natural logarithm was used for Turkish.

It has been observed that increasing the number of iterations does not consistently increase the F1 value for English. As a result, it was decided to have 5 iterations in the conducted tests

which performed as the most appropriate value. 5 iterations are optimal for both Turkish and English as can be seen in Table 5.5 and Table 5.6

# Of Iterations	**F1 Score**
5	**0.5844**
4	0.584
3	0.5843
2	0.5838
1	0.584

Table 5.5 Number of Iterations for English

In the experiment conducted with the Turkish dataset, the highest score was found to be 5, just like in English. Due to the small data set, the F1 score is higher in Turkish tests than in English tests.

# Of Iterations	**F1 Score**
5	**0.7662**
4	0.7656
3	0.7657
2	0.7656
1	0.7649

Table 5.6 Number of Iterations for Turkish

5.3.4. Module Coefficients

As stated in Equation (13), the distributional module and the labeling functions module have coefficients. Tests were made to find the most accurate value. During the tests, the proposed algorithm was run for 1 iteration and extracted 1 pattern. Penalty methods were selected as logarithm for English and natural logarithm for Turkish.

As shown in Table 5.8 and 5.7, the most suitable coefficients for English were selected as 1 for both the distributional module and the labeling functions module.

Coefficient	**F1 Score**
2	0.5837
1	**0.5841**
0.75	0.5836
0.5	0.5839

Table 5.7 Coefficient Value of Distributional Module for English

Coefficient	**F1 Score**
2	0.5838
1	**0.5843**
0.75	0.584
0.5	0.5841

Table 5.8 Coefficient Value of Labeling Functions Module for English

As can be seen in Table 5.9 of the results in the tests with Turkish data, the highest F1 score for the distributional module was obtained with a coefficient of 0.75. On the other hand, labeling functions module contributed positively to the F1 score as its coefficient increased. Looking at Table 5.10, the labeling functions coefficient was chosen 2 in the REPEL comparison experiments.

Coefficient	**F1 Score**
2	0.7615
1	0.7646
0.75	**0.7655**
0.5	0.7649

Table 5.9 Coefficient Value of Distributional Module for Turkish

Coefficient	F1 Score
2	**0.7657**
1	0.7651
0.75	0.7641
0.5	0.7619

Table 5.10 Coefficient Value of Labeling Functions Module for Turkish

5.4. Experiments

5.4.1. Relation Extraction Test

Two experiments were conducted to measure the capabilities of the proposed model. The first test is a multiclass classification test. This test shows the learning ability of pattern module and the entire model. Pattern module that extracts relationships from the text. Other modules are also used to evaluate the excluded candidates. Thus, this test questions the abilities of extracting patterns of pattern module and the learning ability of the entire model at the same time. The proposed model and REPEL work with different score functions in their pattern module. REPEL and the proposed model is trained using the same data and seed instances. The proposed model contains extra "relation-specific" words. To express the effect of the seed instances, the same test was repeated with different numbers of seeds. The results are calculated by using recall, precision and F1 score metrics.

In English experiments, both models ran for 5 iterations and extracted 3 patterns for every iteration. Both Distributional and labeling functions modules coefficients were selected as 1 according to the result of performance analysis tests. Coefficient of Distributional module of REPEL was selected as 1. The proposed model used common logarithm function for English. In Turkish experiments, both models ran for 5 iterations and extracted 10 patterns for every iteration. Coefficient of distributional module of REPEL was selected as 1. The proposed model has coefficients of 0.75 and 2 for distribution and labeling functions modules, respectively. Natural logarithm was used as penalty method for Turkish. The seed

instances in English were translated to Turkish and models used the same seed documents for Turkish and English.

As shown in Table 5.11, the proposed model has better F1 scores than REPEL algorithm. Thus we can say the proposed model showed better performance in detecting relation classes. This difference becomes more evident when the number of samples decreases. As the number of seeds decreased, the difference in accuracy is increased. The first column shows the number of seeds given at the beginning. There are four relationships and these values demonstrate the number of "capitalOf", "founderOf", "nationality" and "childOf" instances respectively. In the dataset from which seed instances were taken, 50 samples could not be found for each relationship. 32 examples were found for the "founderOf" relation and 40 for the "childOf" relation. The number of instances is increased by ten in each test.

# Of Seeds	**REPEL**	**Proposed Model**
50,32,50,40	0.53	**0.57**
40,32,40,40	0.45	**0.59**
30,30,30,30	0.42	**0.58**
20,20,20,20	0.38	**0.57**
10,10,10,10	0.21	**0.50**
0,0,0,0	0.03	**0.47**

Table 5.11 Accuracy Results of Relation Extraction Experiment for English

Table 5.12 presents the classification report for the test of RE in Table 5.11 with 50,32,50,40 seeds. As mentioned previously, in the dataset from which seed instances were taken, 50 samples could not be found for each relationship. 32 examples were found for the "founderOf" relation and 40 for the "childOf" relationship. The number of instances is increased by ten in each test.

Table 5.13 lists the predictions of the proposed model using the confusion matrix. Likewise, Table 5.15 and 5.14 show the results of REPEL. In Table 5.12 and 5.14, models are compared according to their recall, precision and F1 score values. Support column indicates the number of relationship instances. Pattern module extracts candidate patterns with respect to

the instances in the seed document. Pattern module tends to have high precision because candidate patterns with the same named entities are more likely to indicate the same relationship. On the other hand, it tends to have a low recall value because it is difficult to detect the entities and patterns that are not mentioned in the seed document. distributional module seems to have a high recall but low precision values, as it statistically embeds the meanings of words by looking at the entire text.

	precision	***recall***	***f1-score***	***support***
/location/country/capital	0.88	0.84	0.86	5616
/business/company/founders	0.98	0.29	0.44	5752
/people/person/nationality	0.95	0.58	0.72	600
/people/person/children	0.07	0.99	0.13	341

Table 5.12 Classification Report of the Proposed Model

		Predicted Classes			
		capital	founder	nation	child
Actual Classes	capital	4720	29	3	864
	founder	645	1650	15	3442
	nationality	6	1	351	242
	child	2	0	1	338

Table 5.13 Confusion Matrix of the Proposed Model

When Table 5.14 is examined, REPEL generally has higher precision values. As explained before, the common logarithm function in the proposed model is used to encourage the model to find unseen patterns and instances. Thus, the common logarithm function gives lower scores to patterns with few samples, but it may cause decreased scores as seen in the results. Due to the use of a pre-trained model, recall value is higher in the proposed model.

Looking at F1 score, the proposed model is generally more successful than REPEL. Pattern, distributional and labeling function modules run together more successfully than REPEL.

	precision	***recall***	***f1-score***	***support***
/location/country/capital	0.97	0.74	0.84	5616
/business/company/founders	0.98	0.29	0.45	5752
/people/person/nationality	0.99	0.48	0.65	600
/people/person/children	0.06	0.99	0.11	341

Table 5.14 Classification Report of REPEL

		Predicted Classes			
		capital	founder	nation	child
Actual Classes	capital	4183	27	0	1406
	founder	119	1665	1	3967
	nationality	0	1	288	311
	child	3	1	1	336

Table 5.15 Confusion Matrix of REPEL

RE experiments applied in Turkish too. In order to keep the experimental environment constant, two models were run in 5 cycles each cycle to extract 10 patterns for all relations. As a result, the proposed model was more successful when looking at the F1 scores. REPEL has shown low success in experiments as can be seen in Table 5.16. The stability of the Turkish language model and other tools used may have affected the results.

5.4.2. Knowledge Base Completion Test

The second experiment is KBC test. Distributional module of the models measures the semantic similarity between words. A KBC experiment was performed to demonstrate that

# Of Seeds	REPEL	Proposed Model
50,32,50,40	0.0058	**0.7659**
40,32,40,40	0.0058	**0.7658**
30,30,30,30	0.0058	**0.7657**
20,20,20,20	0.0058	**0.7647**
10,10,10,10	0.0058	**0.7651**
0,0,0,0	0.0058	**0.7647**

Table 5.16 Accuracy Results of Relation Extraction Test for Turkish

pre-trained Word2Vec model in the proposed model could make more accurate inferences than REPEL's trained model from scratch. Both models were asked to guess the country by giving capital names and vice versa. Word2Vec returns the word that has the closest word embedding calculated by using the cosine similarity equation to the given word. It is checked whether the word brought is the correct word. If it is not the correct answer, the next word has the closest word embedding checked. Results are calculated by how many attempts the correct answer is reached. The number of Average Guess is used as evaluation metric. For instance, when "ankara" is given to the model, it is expected that model to return "turkey" as result.

In total, 128 capital and country names were used in this study. In Turkish experiments, the same capit and country names are used. They translated to Turkish.

The results in Table 5.17 and Table 5.18 showed that the proposed model's pre-trained Word2Vec model can find the correct answers faster than REPEL's. The proposed model has less number of average guesses.

The same parameters in the Relation extraction experiments are also used here as well. For the English experiments, both models worked 5 cycles and produced 3 patterns in each cycle. Distribution module coefficient is set to 1 and labeling function module coefficient of the proposed model was chosen as 1. The commob logarithm function was used as the penalty method in English experiments. In the tests for Turkish, the models worked in 5 cycles

and produced 10 patterns in each cycle. The distributional module coefficient is set to 1 for REPEL and 0.75 for the proposed model. Natural logarithm was used as the penalty method in Turkish experiments. The same seed documents are used in both Turkish and English.

As the results show in Table 5.17, the proposed model can guess the correct answer with fewer guesses than REPEL. As the number of seeds increased, the number of trials decreased in both models. Nevertheless, the proposed model still had fewer guesses. The first column represents the number of seeds given at the beginning. There are four relationships and these values demonstrate the number of ”capitalOf”, ”founderOf”, ”nationality” and ”childOf” instances respectively. In the dataset from which seed instances were taken, 50 samples could not be found for each relationship. 32 examples were found for the ”founderOf” relation and 40 for the ”childOf” relationship. The number of instances is increased by ten in each test.

# Of Seeds	**REPEL**	**Proposed Model**
50,32,50,40	40.74	**38.74**
40,32,40,40	40.05	**32.82**
30,30,30,30	41.04	**39.64**
20,20,20,20	42.09	**42.02**
10,10,10,10	40.87	**39.87**
0,0,0,0	42.10	**41.88**

Table 5.17 Results of Knowledge Base Completion Test for English

As can be seen in Table 5.18, the proposed model had a much fewer average number of guesses than REPEL. Thus we can say that the proposed model was more successful than REPEL in the KBC test for Turkish. REPEL did not show very successful results in Turkish experiments. Among the reasons for this may be the effect of the Turkish tools used. As the number of seed instances increased, the proposed model found the correct answer with fewer guesses.

In Table 5.19, there are some extracted instances and patterns are shown. Some of the patterns do not have a meaning but generally, the extracted instances are true. In Table 5.19 only one instance per relation is shown. But there is more than one instance for most of the patterns.

# Of Seeds	REPEL	Proposed Model
50,32,50,40	123.96	**44.7**
40,32,40,40	123.96	**46.7**
30,30,30,30	123.96	**46.73**
20,20,20,20	123.96	**46.2**
10,10,10,10	123.96	**47.61**
0,0,0,0	123.96	**48.07**

Table 5.18 Results of Knowledge Base Completion Test for Turkish

Relation	Pattern	Instance
location/country/capital	X based in capital Y	afghanistan, kabul
	X capital Y considered	somalia, mogadishu
	X absorbed in Y vilayat	iraq, baghdad
/business/company/founders	X co founder Y	alexis_ohanian, reddit
	X founder of Y	bill_gates, microsoft
	X founder Y	amazon, jeff_bezos
/people/person/nationality	X decline taking from ruler Y	zimbabwe, robert_mugabe
/people/person/children	X Y children	ptolemy_vi_philometor, cleopatra
	X Y father	ignatius_loyola, society_of_jesus
	X Y protégé son of	adonis_creed, rocky_balboa

Table 5.19 Extracted Patterns and Instances for English

Table 5.20 shows some extracted patterns and NE pairs. When the table is examined, it is seen that generally extracted patterns and instances have meanings. There is no pattern of ”nationality” in the table because a meaningful pattern for the ”nationality” was not extracted by the model.

Relation	Pattern	Instance
location/country/capital	X Y başkenti şehridir	moskova, rusya
	X Y başkenti	moskova, sovyetler_birliği
	X başkenti Y	güney_afrika_cumhuriyeti, pretoria
/business/company/founders	X kurucusu Y	türkiye, mustafa_kemal_atatürk
	X Y kurucusu	karl_liebknecht, alman_sosyal_demokrat_partisi
	X kuran Y	altın_orda_devleti, cuci
/people/person/children	X oğlu Y	yakup, yusuf
	X Y oğlu	cuci, cengiz_han

Table 5.20 Extracted Patterns and Instances for Turkish

6. CONCLUSION

In this thesis, a weakly supervised relation extraction model is proposed. The proposed model, which consists of three main modules, determines whether there is a relationship between named entities by examining the seed information. The proposed model uses weak supervision and data programming together. In order to prove the success of the proposed model, some tests comparing the proposed model and REPEL were conducted. The results show that the model can learn better with fewer resources than REPEL. It has been observed that the proposed model can work with a small number of seeds and can perform relation extraction even in cases where there is no training data.

In the future, it will be aimed to change the pattern module by using a neural network structure. Thus, a deeper syntactic analysis will be possible. It is planned to add BERT[78] as an additional discriminator. Unlike the modules used in the proposed model, BERT examines the sentence as a whole. If it is added to the model, it can increase the learning ability of the model and it will bring a new perspective. It is also considered to update labeling functions at the end of each iteration. Thus, the effectiveness of labeling functions can be increased.

REFERENCES

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

[2] Meng Qu, Xiang Ren, Yu Zhang, and Jiawei Han. Weakly-supervised relation extraction by pattern-enhanced embedding learning. In *Proceedings of the 2018 World Wide Web Conference*, pages 1257–1266. **2018**.

[3] Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pages 8127–8136. PMLR, **2020**.

[4] Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86. **1999**.

[5] Wee Hyong Tok, Amit Bahree, and Senja Filipi. *Practical Weak Supervision*. " O'Reilly Media, Inc.", **2021**.

[6] Burr Settles. Active learning literature survey. **2009**.

[7] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23, **2010**.

[8] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, **2018**.

[9] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. **2008**.

[10] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. **2004**.

[11] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):1–23, **2015**.

[12] David J Miller and Hasan Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. *Advances in neural information processing systems*, 9, **1996**.

[13] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. **2001**.

[14] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. In *Proceedings of the 23rd international conference on Machine learning*, pages 985–992. **2006**.

[15] Richard Zemel and Miguel Carreira-Perpiñán. Proximity graphs for clustering and manifold learning. *Advances in neural information processing systems*, 17, **2004**.

[16] Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209. **1999**.

[17] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR, **2005**.

[18] Yu-Feng Li, Ivor W Tsang, James T Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *Journal of Machine Learning Research*, 14(7), **2013**.

[19] Zhi-Hua Zhou. When semi-supervised learning meets ensemble learning. *Frontiers of Electrical and Electronic Engineering in China*, 6:6–16, **2011**.

[20] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, **2012**.

[21] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, **1997**.

[22] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, **2013**.

[23] Fabrice Muhlenbach, Stéphane Lallich, and Djamel A Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, **2004**.

[24] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. **2008**.

[25] Youngmin Park, Sangwoo Kang, and Jungyun Seo. Information extraction using distant supervision and semantic similarities. *Advances in Electrical and Computer Engineering*, 16(1):11–19, **2016**.

[26] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical natural language processing: a comprehensive guide to building real-world NLP systems*. O'Reilly Media, **2020**.

[27] Kiran Adnan and Rehan Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771, **2019**.

[28] Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*, **2017**.

[29] N Bach and S Badaskar. A review of relation extraction. literature review for language and statistics ii. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 541–550. **2007**.

[30] Kartik Detroja, CK Bhensdadia, and Brijesh S Bhatt. A survey on relation extraction. *Intelligent Systems with Applications*, page 200244, **2023**.

[31] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545. **2011**.

[32] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, **2017**.

[33] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, **2016**.

[34] Ekaterina Kochmar. Getting started with natural language processing. *(No Title)*, **2022**.

[35] Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda. *Applied text analysis with Python: Enabling language-aware data products with machine learning*. " O'Reilly Media, Inc.", **2018**.

[36] Alisa Smirnova and Philippe Cudré-Mauroux. Relation extraction using distant supervision: A survey. *ACM Computing Surveys (CSUR)*, 51(5):1–35, **2018**.

[37] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge*

Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21, pages 148–163. Springer, **2010**.

[38] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 541–550. **2011**.

[39] Eugene Agichtein and Luis Gravano. Extracting relations from large plain-text collections. **1999**.

[40] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. **2012**.

[41] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782. **2013**.

[42] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 74–84. **2013**.

[43] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y Chang. Errata: Distant supervision for relation extraction with matrix completion. *arXiv preprint arXiv:1411.4455*, **2014**.

[44] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1456–1466. **2011**.

[45] Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 54–59. **2012**.

[46] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. Relation extraction with relation topics. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436. **2011**.

[47] Benjamin Roth and Dietrich Klakow. Combining generative and discriminative model scores for distant supervision. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 24–29. **2013**.

[48] Tapas Nayak, Navonil Majumder, Pawan Goyal, and Soujanya Poria. Deep neural approaches to relation triplets extraction: A comprehensive survey. *Cognitive Computation*, 13:1215–1232, **2021**.

[49] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011. **2009**.

[50] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–729. **2012**.

[51] Johannes Kirschnick, Alan Akbik, and Holmer Hemsen. Freepal: A large collection of deep lexico-syntactic patterns for relation extraction. In *LREC*, pages 2071–2075. **2014**.

[52] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1219–1228. **2014**.

[53] Desh Raj, Sunil Sahu, and Ashish Anand. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)*, pages 311–321. **2017**.

[54] Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Incorporating relation paths in neural relation extraction. *arXiv preprint arXiv:1609.07479*, **2016**.

[55] Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245. **2018**.

[56] Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*, **2018**.

[57] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133. **2016**.

[58] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344. **2014**.

[59] Hongtao Lin, Jun Yan, Meng Qu, and Xiang Ren. Learning dual retrieval module for semi-supervised relation extraction. In *The World Wide Web Conference*, pages 1073–1083. **2019**.

[60] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480. **2016**.

[61] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1556–1567. **2014**.

[62] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 732–738. **2014**.

[63] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Distantly supervised web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, **2016**.

[64] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S Weld. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901. **2014**.

[65] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*, pages 640–656. Springer, **2013**.

[66] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. **2008**.

[67] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 conference of the north American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129. **2015**.

[68] Xianpei Han and Le Sun. Global distant supervision for relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30. **2016**.

[69] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 766–773. IEEE, **2011**.

[70] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. Holoclean: Holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820*, **2017**.

[71] Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pages 31–40. **2008**.

[72] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(2), **2010**.

[73] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. **1998**.

[74] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1165–1174. **2015**.

[75] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, **2013**.

[76] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, **2015**.

[77] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420, **2017**.

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