Heidelberg University
Institute of Computer Science
Database Research Group

Master's Thesis

Open Numerical Information Extraction

Name:	Hasan Shahid Malik
Matriculation Number:	3439403
Supervisor:	Prof. Dr. Michael Gertz
Datum der Abgabe:	March 1, 2020

Ich versichere, dass ich diese Master-Arbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe und die Grundsätze und Empfehlungen “Verantwortung in der Wissenschaft” der Universität Heidelberg beachtet wurden.

Hasan Matile

Abgabedatum: March 1, 2020

Zusammenfassung

Mehr als die halbe Weltbevölkerung hat jüngsten Zahlen zu Folge Zugang zum Internet. Dadurch werden täglich Unmengen an unstrukturierten Texten kreiert. Die Umsetzung von unstrukturiertem Text in strukturierten Text nennt man Informations-Extraktion (IE). In IE beschäftigt man sich mit dem Teilbereich der Extraktion von Beziehungen (Relationen) zwischen zwei oder mehreren Einheiten der natürlichen Sprache. Seit den Anfängen von IE hat man bedeutende Entwicklungen gesehen. Verglichen mit den anfänglichen einfachsten Modellen, die nur ein paar vorselektierte Beziehungen erkannten, gibt es nun sog. Open IE Systeme, die ganz automatisiert ohne Kontrolle neue Relationen erkennen können. Wir studieren in dieser Abhandlung einen kürzlich definierten Teilbereich von IE, der noch nicht so weit entwickelt ist: die numerische Relationsextraktion. Numerische Beziehungsextraktion besteht aus der Aufgabe, semantische Relationen zwischen einer Einheit und einer Menge in unstrukturiertem Text zu extrahieren. Bis jetzt gibt es nur drei numerische Relationsextraktion-Modelle, und Open IE ist nur eines davon. Aus diesem Grunde gibt es bisher keine Studien zu den möglichen Anwendungen der numerischen Relationen.

Wir entwickeln hier in dieser Arbeit das erste regelbasierte Open IE speziell für numerische Relationen und wir forschen an einem möglichen Anwendungsfall der numerischen Relationen im Bereich Themen-Modellierung. Wir entwickeln hier einen innovativen Ansatz mit einer Aneinanderreihung von Substantiven im Rahmen einer Reihe von uns vorgeschlagenen Kriterien, um die für die Relation relevanten Komponenten in einem Satz zu bilden. Nachdem wir Syntax-Muster in einfachen Sätzen studieren, formulieren wir Regeln, die die Einheiten, die Beziehunen und Mengen in einem Satz identifizieren und extrahieren, wobei wir auf die Positionierung der dazugehörigen Komponenten achten. Wir vergleichen unsere Extraktionen mit denen von anderen numerischen Relations-Extraktions-Modellen und erkennen, daß für nicht-lernende Modelle unsere noch den größten Ertrag bringen. Im Vergleich zu dem anderen Open IE Modell in diesem Teilbereich, unser Modell produziert etwa ein Drittel des Ertrages, aber mit höheren qualitativen Extrationen.

Wir schlagen eine innovativen Anwendung der numerischen Relations-Extraktion vor als Mittel zur Dimensions-Reduktion in Themen-Modellierung. In einer Dokumenten-Sammlung können wir jedes Dokument auf seine konstituierende numerische Relation reduzieren. Wir lassen LDA und NMF, zwei beliebte Themen-Modellierungs-Algorithmen, die original- und reduzierte

Sammlung berechnen und vergleichen deren Ergebnisse. Wir stellen fest, daß die Ergebnisse obwohl nicht ganz schlüssig auf mögliche Anwendungsfälle der numerischen Relation im Bereich des Dokumenten-Clustering hinweisen.

Abstract

Recent figures show that today more than half of the human population has access to the internet in one form or another. This has resulted in vast amounts of unstructured text data being generated every day. The task of converting unstructured text into a structured representation is called Information Extraction (IE). One of the domains in IE is relation extraction, which is the task of extracting semantic relations between two or more entities in natural language. Since its inception, the field of relation extraction has seen significant development. From simple models capable of extracting just a few preselected relations, there are now Open IE systems capable of automatically extracting new relations without any supervision. In this thesis, we study a recently defined sub-domain of relation extraction that has not yet seen as much development: Numerical relation extraction. Numerical relation extraction is the task of extracting semantic relations between an entity and a quantity in unstructured text. To date, there are only three numerical relation extraction models, and only one of them is Open IE. As a result of this, there has been no study into the possible uses of numerical relations.

In this thesis we develop the first rule-based Open IE model specifically designed for numerical relations, and we explore a possible use case of numerical relations in the domain of topic modeling. We develop a novel approach of chaining noun phrases together, under a set of criteria that we also propose, to form relation-relevant components in a sentence. By studying syntactic patterns in simple sentences, we form rules capable of identifying and extracting the entity, relation and quantity in a sentence based on the positioning of their associated components. We evaluate our extractions against those produced by other numerical relation extraction models and find that of the non-learning models, ours produces the greatest yield. Compared with the other Open IE model in this sub-domain, our model produces about one third of the yield, but with higher quality extractions.

We propose a novel use of numerical relation extraction as a means of dimension reduction in topic modeling. Given a corpus of documents, we reduce each document to its constituent numerical relations. We compute LDA and NMF, two popular topic modeling algorithms, on both the original and the reduced corpus and compare their results. We find that, while inconclusive, the results hint at possible use cases of numerical relations in the domain of document clustering.

Contents

1 Introduction

Humanity is truly living in the age of data. Recent figures[1] report that as of January 2020, almost 4.54 billion people now have access to the internet. That figure represents more than half of the human population. In 2021 the number of people on social media is projected[2] to grow to near 3.1 billion. With so much of the population online, it is no wonder that humanity is faced with staggering amounts of data, and a large part of that data is in the form of unstructured text. Information extraction (IE) is the task of converting this unstructured, natural language text into a structured representation using Natural Language Processing (NLP) techniques [Jurafsky and Martin, 2009]. One of the key tools in the toolkit of IE is relation extraction, which is the extraction of relational tuples of the form *(arg1, rel, arg2)* consisting of two arguments and a relation phrase connecting them from a sentence [Niklaus et al., 2018]. For example, the simple sentence "The water is warm" can be represented by the relational tuple: *(The water, is, warm)*. Traditional methods approached this task by learning a few syntax patterns for a predetermined list of relations, and applying these patterns on small, homogeneous text data sets in order to extract relation tuples [Agichtein and Gravano, 2000, Brin, 1999]. However, this greatly limited the types of relations that could be identified. In order to identify a new relation, the relation would first need to be explicitly provided to the model alongside labeled training data. The model would then have to incorporate new rules to be able to identify and extract the new relation. These early techniques suffered greatly from lack of scalability, and so a new idea was proposed by [Etzioni et al., 2008] in the form of TEXTRUNNER, which was the first Open Information Extraction (Open IE) model. The goal of an Open IE model is to be able to extract any type of relation it encounters, without a pre-specified relation list. In their paper, [Etzioni et al., 2008] identified the following major challenges for Open IE models:

- **Automation.** Previous systems relied on supervised training sets and predetermined relations. Open IE models must be able to automatically detect and extract new relations in a single pass over a corpus, without needing a predetermined list of relations or an extensive labeled training set. At most it can use a small set

[1] www.statista.com/statistics/617136/digital-population-worldwide/

[2] www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

of seed facts or extraction patterns with which it must automatically learn new relations.

- **Corpus Heterogeneity.** Open IE models should be tested on a text data sets from a variety of domains, because learning models based on syntax parsers and Named Entity Recognition (NER) commonly work well when trained and applied on one domain, but perform poorly when tested on different domains.

- **Efficiency.** Open IE systems must be computationally efficient since they are designed to scale to large amounts of text. Therefore they should avoid using dependency parsers and making large numbers of search engine queries.

Naturally the advance of technology and the development of more efficient algorithms have significantly improved the speed of many dependency parsers; therefore the final criteria may be a little less relevant in this day and age.

This first Open IE model heralded the arrival of many more attempts at creating models capable of automatically learning new relations. In 2017, this resulted in an Open IE model called BONIE [Saha et al., 2017], aimed specifically at extracting a subsection of relations; namely, numerical relations. This work built on a previous study by [Madaan et al., 2016], who defined numerical relation extraction as the task of extracting a binary relation between an entity and a quantity. For example, from the sentence "Aluminium is a chemical element in the boron group with symbol Al and atomic number 13", a numerical relation extractor wants to extract the tuple *(Aluminium, has atomic number, 13).* The work by [Madaan et al., 2016] found that numerical relation extraction has a few peculiarities that are uncommon in the wider field of relation extraction:

- Quantities are often expressed as relative values rather than the actual values.

- Modified and scoped entities are more common.

- Many numerical relations are mediated by a handful of explicitly stated keywords, or at the very least, are alluded to by the units of the quantity.

This sub-domain of relation extraction is still rather new and therefore has only resulted in a few works and only one Open IE model. A consequence of the lack of work in the numerical relation extraction field is that there has been no investigation into the possible uses of extracted numerical relations. The use of numerical relations in text tends to be more reserved than normal relations, but numerical relations themselves can give considerable information. For example, given the following two numerical extractions

(X, has top speed of, 110 kmph) (X, seats, 4 passengers), we can pretty safely guess that X is a car or similar vehicle. It is interesting to consider the applications of numerical relations.

1.1 Objectives and Contributions

In this thesis we aim to help develop the emerging domain of numerical relation extraction, and break its reliance on data intensive models. Additionally we explore a possible use case of the extracted numerical relations as a means of reducing data requirements for topic modeling. This thesis, therefore, makes the following contributions:

- We propose a new approach to removing uninformative clauses from sentences containing numerical relations.
- We introduce a new method of chaining consecutive noun phrases to build entities and relation components.
- We construct a set of grammar based rules to identify the entity and relation components based on their positions.
- We build the first rule-based numerical Open IE model and evaluate its performance against other works in the field of numerical relation extraction.
- We conduct an exploration of the impact that reducing a corpus to its constituent numerical relations has on topic identification.
- Since there are few corpora with labeled numerical relations, we create our own.
- We evaluate the performance of two of the most popular topic modeling algorithms on various iterations of our test set, and compare their performances.

1.2 Overview of the Thesis Structure

This thesis is structured as follows: In Chapter 2, we give a brief overview of the techniques used in NLP followed by an introduction and discussion of related works from the domains of relation extraction, numerical relation extraction, and topic modeling. In Chapter 3 we formally define our numerical relation extraction model and describe how we extend its functionality to explore the use of numerical relations in topic mining. In Chapter 4 we establish the test data sets and the evaluation metrics we use for each of our two tasks, and then present and discuss the results of our experiments. Finally, in Chapter 5 we conclude with a review of our work and preview of possible future work.

2 Basics and Related work

In this chapter we provide an introduction to all the terms and ideas we will use when describing our model. In Section 2.1 we introduce several of the standard NLP processes that are used in most text analysis tasks. This is followed by an introduction to the domain of relation extraction and its related works in Section 2.2. Section 2.3 describes numerical relation extraction, and gives an overview of the literature most relevant to our model. Finally, in Section 2.4 we introduce topic modeling and describe its representative algorithms.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence that attempts to understand human language and quantify it in such a way as to be machine processable. This is no easy task, as human language is very diverse and complex. The operations described in the following subsections are all part of the NLP toolkit for deriving meaning from human language.

2.1.1 Tokenization

Tokenization is the process in which a block of text is split (often on whitespaces) into its constituent atomic units [Habert et al., 1998]. Each unit, be it a word, number or punctuation, is then referred to as a *token*. This task can be more difficult than it seems at first glance; a tokenizor needs to be able to differentiate between punctuation used to delineate a sentence versus punctuation used in an abbreviation or to write a decimal number. Tokenization is an essential part of data preparation for virtually every text analysis task and is often a preliminary requirement for other text cleaning procedures.

2.1.2 Stop Words

One such text cleaning procedure is *stop word* removal, which is the removal of the most common and typically least domain distinguishing words from a given text. Words such as “a”, “of” and “for” fall into this category known as stop words; their removal is predicated on the idea that they tend to be among the least informative but most highly represented words in a text, and thus can assert disproportionate weight (versus their informative value) on any learning algorithm applied on the text. However, stop words are domain and context dependent; for example, the word “I” in the sentence “I work at Apple” may be considered a stop word in a topic mining task, however, “I” represents one of the entities in the relation“employee of”. As such, care must be taken when selecting stop words. [Manning and Schütze, 1999]

2.1.3 Lemmatization

Natural language is not composed solely of the root form of words, that is, the dictionary definition form of words; grammar dictates that words take different forms depending on their context. For example, the words “walk”, “walks”, “walked” and “walking” are all conjugations of the same root form “walk”. In nlp tasks such as topic mining, it does not matter what form the word appears in, just that it appears at all. In such cases lemmatization is used to reduce words to their root forms, which are known as *lemmas*. [Manning et al., 2008, Manning and Schütze, 1999]

2.1.4 Part-of-Speech Tagging

A word’s part-of-speech (POS) is the lexical category to which the word is assigned based on its syntactic function in a sentence. These lexical categories include noun, verb, adjective, preposition and so on. Part-of-Speech tagging is the process of assigning tokens to their respective POS categories. The POS tags can be very informative; they are used for *noun phrase chunking*, which is the process by which a sentence is divided into non-overlapping noun phrases; they are used in *named entity recognition*, which is the information extraction task of detecting mentions to members of pre-defined classes such as persons, organizations and locations in free text [Jurafsky and Martin, 2009]; and they are also used in topic mining and keyword extraction tasks where sometimes only particular POS tags (e.g. nouns) are considered as possible keywords. [Mahata et al., 2018]

2.1.5 Dependency parsing

In the sentence “I threw the ball at the wall”, the word “I” is the noun subject of the verb “threw”, the noun phrase “the ball” is the direct object of the verb, and the noun phrase “the wall” is the object of the preposition “at”, which, in turn, is a prepositional modifier of “threw” (see Figure 2.1). Revealing the syntactic structure of a sentence, and the links between its words, is the task of dependency parsing and is the job of a dependency parser. The labels *nsubj* (noun subject), *dobj* (direct object), *pobj* (object of a preposition) and *prep* (prepositional modifier) are all examples of dependency tags which describe the directed link to a word from its syntactic "parent" that the word modifies. The syntactic structure of a sentence takes on a tree shape, with the *root* tag being assigned to the word at the head of the tree, i.e. the word with no syntactic parent. In most cases, the finite verb of a sentence is taken to be the root, and all dependencies stem from it [Hudson, 2010, Jurafsky and Martin, 2009]. Every word in a sentence, save the root word, has a syntactic parent; the word itself being one the syntactic children of its syntactic parent. Figure 2.2 shows the dependency tree of the example sentence above. Dependency parsing plays a substantial role in relation extraction tasks as shown in [Madaan et al., 2016].

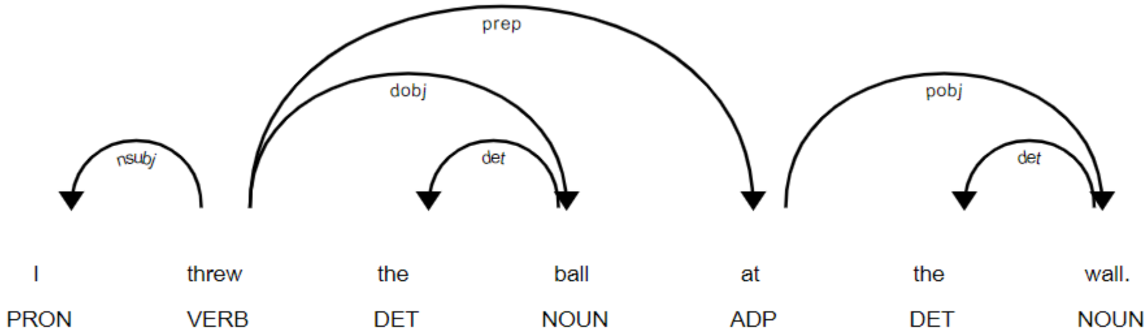


Figure 2.1: Dependency parse of the sentence "I threw the ball at the wall".

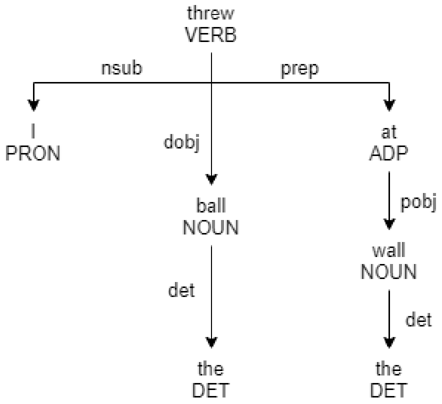


Figure 2.2: Dependency tree of the sentence "I threw the ball at the wall". Arrows point from syntactic parent to syntactic child, with the verb "threw" as the *root* of the sentence.

2.1.6 Coreference resolution

Oftentimes a text will mention an entity once and later refer to it using a pronoun. For example, in the text "I had a dog. It barked a lot." the word "It" clearly refers to "dog" and not "I". Coreference resolution is the task of finding and linking all mentions that refer to the same entity in a text. This can be very difficult because language can be highly ambiguous. It can require a great deal of background knowledge in order to figure out what entity a phrase is referring to. For example, to understand the sentence "John rocked Billy in his cradle. The new father was tired of the baby's restlessness", a computer would need to be able to link "cradle" with "baby", and therefore understand that "Billy in his cradle" is "the baby", and that a baby cannot be a "new father", hence "John" must be "the new father". [Jurafsky and Martin, 2009]

2.2 Relation Extraction

Relation Extraction is the task of the detection and identification of semantic relationships between two or more arguments in a text. These arguments can be entities, such as persons, organizations and locations; objects; or other attributes like color or shape [Zelenko et al., 2002, Madaan et al., 2016]. The history of this domain goes back to the Message Understanding Conferences held between 1989 and 1997 [Jung et al., 2012]. Since then, most of the approaches developed to tackle this task can be classified into four main categories:

- Supervised approaches in which contextual features are selected from a sentence and used to build a feature vector that is then classified into a predefined relation category [Kambhatla, 2004]. The problem with fully supervised approaches, however, is their requirement for vast amounts of labeled data per relation learned. This makes such an approach time consuming to scale up to more relations.

- Semi-supervised learning methods get around this problem by automatically labeling an unsupervised corpus using a knowledge base of facts in a process known as *distant supervision*. The work by [Mintz et al., 2009] uses distant supervision alongside a feature based system to train a classifier to learn a relation between two entities. MultiR [Hoffmann et al., 2011] uses a graphical model that utilizes the features described by [Mintz et al., 2009] to detect and identify multiple relations between the same two entities.

- Bootstrapping methods only require a handful of manually selected lexical patterns, known as *seeds*, which are then matched on unlabeled data to find similar terms and learn their patterns and recursively build up the number of recognized extraction patterns.[Waegel, 2003]

- Kernel-based methods often parse input texts into tree-like structures using POS tagging and dependency parsing, and then define kernels that compute the similarity of the relation defining components between instances of input text. High similarity indicates that the texts share the same relation, and can thus be classified together. [Zelenko et al., 2002, Jung et al., 2012]

Open Information Extraction (Open IE) systems extract relations between entities without requiring a pre-specified vocabulary. These systems often use distant supervision or bootstrapping methods to get around the requirement for large amount of labelled data. The first open IE model was TEXTRUNNER [Etzioni et al., 2008] which analyses the words between each pair of noun phrases in an input sentence. If the interposed words are classified as a relation based on the presence of the appropriate features (e.g. matches a sequence of POS tags, has a certain number of words, etc.) then the triple consisting of the pair of noun phrases (the entities) and the words between them (the relation) is output. A later work by [Fader et al., 2011] made the observation that TEXTRUNNER often outputs incoherent or trivial relations when non-trivial relations are present. For example, in the sentence "Abbey Road is an album by The Beatles" TEXTRUNNER outputs the triple *(Abbey Road, is, an album)*, rather than *(Abbey Road, is an album by, The Beatles)*. [Fader et al., 2011] improved on this in their rule-based open IE model called REVERB, which imposes syntactic constraints on the relations by requiring that the relation phrase consist, if possible, of a verb phrase followed by a noun phrase followed by a preposition. In REVERB, the relation phrase is determined first, and then the entities are identified based on their positions around the relation.
The relation extraction field has seen significant advancement in the past two decades. However, not all subfields have received as much attention.

2.3 Numerical relation extraction

Numerical relation extraction is a subfield of relation extraction that deals with the extraction of a numerical relation between an entity and a quantity. Such relations can include more recognized examples such as "length" and "weight", as well as more general examples such as "population", "inflation rate" and "GDP". While there has been work done on extracting quantities from text, comparatively little has been done on identifying numerical relations. This is a shame, because numerical relation extraction poses a different set of challenges than standard relation extraction.

- Distant supervision methods are based on the idea that if two entities from a relation knowledge base (KB) occur together in a sentence, then that sentence contains the associated relation from the KB. However, quantities can occur in far more contexts than entities can. For example, the relation between the word “Germany” and the number 1 could be any number of things from a sports score to the illiteracy rate. Additionally, quantities in the KB might be reported to a different degree of accuracy, or with different units than the quantities in the corpus, thereby making distant supervision considerably more difficult.

- Quantities, particularly in financial contexts, are often expressed in terms of their relative value rather than the actual value. This happens all the time with stock market news such as “FTSE 100 closes down 82 points” and makes defining the relation difficult.

- Modified and scoped entities are more common since numerical relations can be easily re-scoped to different aspects of the entity. For example “the population of Germany” versus “the population of eastern Germany”.

Fortunately, however, many numerical relations are mediated by a handful of keywords or phrases that are often explicitly stated in the sentence. An example of this is the phrase “interest rate” in the sentence “the interest rate of Germany is 2%”. Detecting and identifying these keywords and phrases can simplify the numerical extraction problem. The first works in the field of numerical relation extraction with express purpose of extracting general numerical relations such as population, life expectancy and interest rates were NumberRule and NumberTron [Madaan et al., 2016], while BONIE [Saha et al., 2017] was the first numerical Open IE model attempted in the domain.

2.3.1 NumberRule

NumberRule is a rule-based, non-learning, numerical relation extractor that takes a sentence and list of keywords per relation as input. It uses Named Entity Recognition to identify candidate entities in a sentence and then finds the shortest path in the dependency parse between a candidate entity and a quantity, looking for relation keywords either on, or immediately connected to, the shortest path. If a keyword is found, then the associated relation is returned from the list. NumberRule applies four tests to the extraction before outputting it:

1. The measurement units of the quantity must match those expected by the relation.

2. There can be no change words in the sentence; the quantity must express the actual value rather than a relative value.
3. The entity must not be scoped or otherwise modified.
4. The keywords must not be scoped or otherwise modified.

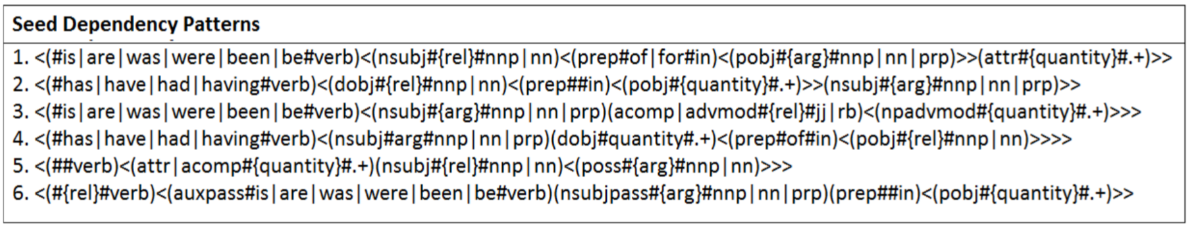
If any of these tests fail, the extraction is rejected, and the next candidate entity is considered. If all the tests succeed, the extraction is output in the form: *relation(entity, quantity).* This selectivity results in fairly poor recall in free text.

2.3.2 NumberTron

NumberTron is a distantly supervised probabilistic graphical model inspired by MultiR [Hoffmann et al., 2011]. It uses the same keywords per relation list that NumberRule does, but can learn new keywords during the training phase. In order to create training data, NumberTron aligns an unlabeled corpus with a knowledge base of facts, using partial matching on normalized quantities in order to match entity and quantity instances in the corpus to entity and quantity instances in the KB. It then uses the lexical and syntactic features described in [Mintz et al., 2009] in a perceptron algorithm to learn extraction patterns and new keywords. Extractions are once again output in the form *relation(entity, quantity)* where the relation is one of the predefined relations, the entity is unmodified, and the triplet of *(entity, relation, quantity)* is corroborated by the KB of facts. Both NumberRule and NumberTron are tested on 425 sentences in the domain of geopolitical relations, and therefore both the KB and the predefined list of relations are of that domain.

2.3.3 BONIE

BONIE is an open IE model specifically built to handle numerical relation extractions; like all open IE models, BONIE does not require a pre-specified list of relations. The inputs to BONIE are a large corpus (roughly 2 million sentences) of unlabeled data, and a set of six manually selected high-precision dependency patterns (see Figure 2.3) which are matched against the corpus to generate the seed facts that are used for bootstrapping. Quantities in the corpus and the seed facts are normalized. BONIE then finds sentences in the corpus that contain all the words in a particular seed fact and, if found, creates *(sentence, seed fact)* pairs for all such sentences across all seeds. The entity, relation and quantity components of the sentence are known from the seed fact, and so they are extracted from the sentence along with any modifying or scoping words that are



<table>
<tr><th>Seed Dependency Patterns</th></tr>
<tr><td>1. <(#is|are|was|were|been|be#verb)<(nsubj#{rel}#nnp|nn)<(prep#of|for#in)<(pobj#{arg}#nnp|nn|prp)>>(attr#{quantity}#.+)>></td></tr>
<tr><td>2. <(#has|have|had|having#verb)<(dobj#{rel}#nnp|nn)<(prep##in)<(pobj#{quantity}#.+)>>(nsubj#{arg}#nnp|nn|prp)>></td></tr>
<tr><td>3. <(#is|are|was|were|been|be#verb)<(nsubj#{arg}#nnp|nn|prp)(acomp|advmod#{rel}#jj|rb)<(npadvmod#{quantity}#.+)>>></td></tr>
<tr><td>4. <(#has|have|had|having#verb)<(nsubj#arg#nnp|nn|prp)(dobj#quantity#.+)<(prep#of#in)<(pobj#{rel}#nnp|nn)>>>></td></tr>
<tr><td>5. <(##verb)<(attr|acomp#{quantity}#.+)(nsubj#{rel}#nnp|nn)<(poss#{arg}#nnp|nn)>>></td></tr>
<tr><td>6. <(#{rel}#verb)<(auxpass#is|are|was|were|been|be#verb)(nsubjpass#{arg}#nnp|nn|prp)(prep##in)<(pobj#{quantity}#.+)>></td></tr>
</table>

Figure 2.3: BONIE seed patterns. Each word is encoded as $< depLabel\#word\#POSTag >$, where *depLabel* is the dependency tag of the token, *word* is the word at the token, *POSTag* is its part of speech tag; # is a delimiter separating them. {*rel*}, {*arg*} and {*quantity*} in the patterns are placeholders for the relation, argument and quantity phrase syntactic parent tokens respectively.

part of the same phrase. If the syntactic parent word of the relation component is an adjective or an adverb, then BONIE looks this word up in WordNet[1], which is an online English lexical database, to get the word's noun form and append it to the relation. If the quantity is a unit-less value then the phrase "Number of" is appended to the relation. BONIE then outputs this numerical relation as a triple of the form: *(entity, relation, quantity)*. The original sentence is then parsed into dependency patterns with the entity, relation and quantity components replaced by placeholders. The minimal subtree dependency pattern containing the entity, relation and quantity placeholders is then used to find more facts in the corpus.

2.4 Topic Modelling

Topic modeling is the unsupervised machine learning task of detecting latent topics in documents, where each document is a mixture of topics and each topic is a probability distribution over words [Silge and Robinson, 2017]. The prevailing idea behind many topic modeling algorithms is that different topics contain different words with higher or lower probability. For instance, a document containing the words "cat" and "dog" several times suggests it is a passage about pets; intuitively, for such a document to contain the word "subatomic" would be rather unlikely.
There are two distinctive topic learning approaches that have developed in the domain of topic modeling [Chen et al., 2019]:

- Probabilistic models iteratively try to find the probabilities $p(word|topic)$ and $p(topic|document)$ for each word in each document after initial random assignment until convergence based on algorithm specific criteria. The representative work of this approach is Latent Dirichlet Allocation (LDA) [Blei et al., 2003] which tries

[1] https://wordnet.princeton.edu/

to infer the topics in a corpus by reverse engineering the generation process of the corpus. LDA treats documents as random mixtures of topics where each topic is a distribution over all the words in the corpus, and tries to calculate the parameters of these distributions using the observed distribution of words across the documents in the corpus.

- Linear algebraic models try to factorize a high dimensional matrices into lower dimensional matrices. The representative work of this approach is Non-negative Matrix Factorization (NMF) that attempts to factorize the document-term matrix of a corpus into two lower dimensional non-negative matrices, namely, the topic-word matrix and the document-topic matrix.

Any form of clustering operations on text face certain issues. Any naive text data representation has a very high dimensionality with sparse underlying data. This is because the size of a corpus vocabulary is generally far greater than the number of words in any one text in that corpus; if we construct a document-term matrix of such a corpus, it will have many columns (corresponding to words) and will largely be empty. Additionally, words are often semantically related to one another. If two words have the same meaning and thus can be used interchangeably in a text, they may never co-occur; this can mislead clustering attempts [Allahyari et al., 2017]. Given how little work has been done in the field of numerical relation extraction, it is not surprising that we are unable to find any previous works that study the usefulness of numerical relations in topic modeling.

In the next chapter we construct our numerical relation extraction model and its extension into topic modeling using the concepts we introduce in Chapter 2.

3 Modeling Numerical Relations

In this chapter we establish our numerical relation extraction model and its subsequent use in topic modeling. In Section 3.1 we give a comprehensive overview of the model. In Section 3.2, we present the rules that our model uses to perform extractions. In Sections 3.3 though 3.6 we formally define the preprocessing and extraction phases of our model. Lastly, Section 3.7 formally defines the LDA and NMF algorithms we use for topic modeling.

3.1 Objectives and Overview

As we saw in Section 2.3, there have been relatively few works specifically aimed at numerical relation extraction; for this reason, many possibilities have gone unexplored. The greater field of relation extraction faces the difficulty that any two entities in a sentence may be related, so detecting which entities are being referred to adds a layer of complexity. The relations themselves may be entirely implicit and only discoverable through a background knowledge base. Numerical relations on the other hand, while being more difficult to match against a KB of facts, are easier to extract since one of the arguments (the quantity) is already known. Sentences containing numerical relations are often formed in such a way that makes the the relation explicit or at the very least identifiable by a keyword such as “tall” for the relation “height of”, or a measurement unit such as “meters”. It should be much easier to construct a extractor that builds relations through the existing words in the sentence, without requiring a prespecified list of relations, for numerical relations versus the broader field of entity-entity relations. BONIE [Saha et al., 2017] is the only numerical relation extractor that explicitly sets out to do this task. However, since BONIE uses bootstrapping to expand its extraction pattern list, it requires vast amounts of text data which comes with its own set of problems. Any supervised or semi-supervised learning approach will require a lot of labeled training data, and the extractor will be limited to the domain on which it is trained. The evidence for this is the NumberTron [Madaan et al., 2016] model described in Section 2.3. Since NumberTron is a distantly supervised model, its extractions are limited by the knowledge base used to label the training data. NumberTron was provided a predetermined list relations to identify, and while it could learn new keywords for identifying those relations, it could not extract relations not in the list. In order to be able to carry out sentence level extractions without the need for large amounts of data, we must turn to a rule based system. A rule-based extractor does not need to be as restrictive about the input data as the NumberRule [Madaan et al., 2016] model might suggest. NumberRule was designed explicitly to extract only a handful of predefined relations that could later be compared to NumberTron's extractions. As a byproduct of this, NumberRule was also unable to deal with entity or relation scoping since a knowledge base is far more likely to contain un-modified relations between un-scoped entities and absolute values, such as *(India, interest rate, 3%)*, than it is to accommodate the innumerable relations between scoped entities and relative values, such as *(rural India, literacy rate lower than urban by, 20%)*. Additionally, NumberRule and NumberTron use Named Entity Recognition (NER) to identify the entity. Most available named entity recognition model can only identify a select few entities such as famous people, locations and organizations; any extensions to this require a large amount of labeled training data. This is fine for NumberRule because it was tested on country names; however this is not sufficient for our model since the vast majority of free text contains few recognizable entities [Etzioni et al., 2008].

We construct a rule-based numerical relation extractor capable of constructing grammatically coherent relations through the words stated in the input text, with extra relation specific words only added if they are not explicitly stated. Our model should rely solely on the syntactic structure of sentences to extract the entity, and not on NER. Additionally our model should be capable of processing text containing scoped entities and relative values, and outputting appropriately modified relations.

3.1.1 Overview of Numerical Relation Extraction

Our model takes document as input, tokenizes the words, annotates it with POS tags and dependency tags and then splits it into its constituent sentences, removing any sentences that do not contain quantities. For each remaining sentence, our model creates a representation of the sentence that allows us to remove comma-separated clauses that are not relevant to our extraction. We deem a clause to not be relevant to an extraction if it does not contain the quantity or the most closely related syntactic parent verb of the quantity that has a noun subject. The reason behind this is that if the quantity is the noun subject of the verb, then the entity appears in the same clause whether as an independent noun phrase, or as part of the quantity phrase as in the sentence "200

troops march into battle". If the quantity is not the noun subject of the verb, then either a relation keyword or the entity itself is the noun subject of the verb. NumberRule and NumberTron identify the entity first using NER, and then uses the shortest dependency parse between the quantity and the entity to look for relevant information. In our method, we remove extraneous words before setting out to identify the entity since we do not use NER. We study the syntactic structures of a number of short sentences containing quantities, including some of BONIE's seed facts, and attempt to map out the position of the entity of each sentence relative to the verb and quantity with a view to generalizing any patterns we find to more complex sentences. Given a sentence, we chain consecutive noun phrases together into components and then use the positioning of the components and the generalized patterns to identify which is the entity and which is the relation. Our model shares some similarities with the REVERB model [Fader et al., 2011] in that the entities (or entity in our case) are discovered based on their positioning with respect to the syntactic parent verb of the clause. While REVERB extracts both the preceding and the following noun phrases, we only need to extract one or the other depending on the position of the quantity with respect to the verb. We then attempt to identify any implicit relation in the text based on the explicit relation-relevant keywords contained in the relation component. This is a two step process: First we check the units of the quantity; if they are a recognized unit of measure then we look them up in a KB that returns their associated measure. For example, if the units of the quantity are "meters" or "cm" or "miles" then the associated measure returned is "length". In the second step, this measure is compared against a second KB which can optionally be provided as an input, which further subdivides measures depending on the presence of certain keywords. For example, if the relation component consists of the words "above sea level" and the unit is "meters", then we know the measure is of "height" rather than "length". This second KB allows us to fine tune the wording of the extracted measure. This measure, if not already present, is then included in the relation component of the extraction. Note, however, that these two knowledge bases only provide additional information; they are in no way a requirement for the extraction to run, making our model an Open IE system despite being rule-based. The relation extraction phase concludes with outputting a tuple of the form *(entity, relation, quantity)*. Figure 3.1 shows the entire numerical relation extraction pipeline. The extractions generated then become the basis for the second part of our model: Using numerical relations as a means of dimension reduction in topic modeling.

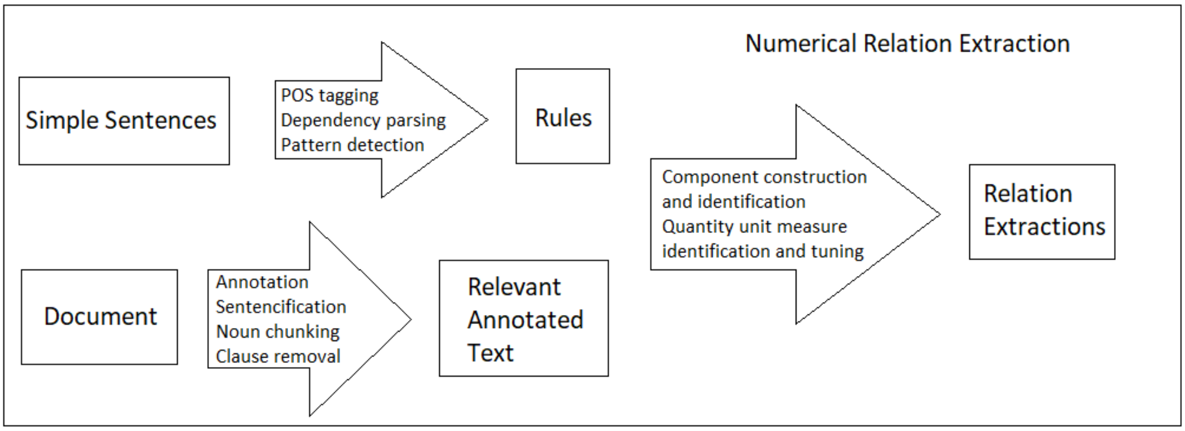


Figure 3.1: Our rule-based, numerical relation extractor's processing pipeline.

3.1.2 Overview of Topic Modeling

As we saw in Section 2.4, textual data suffers from high dimensionality. This is a natural byproduct of the vastness of many human language vocabularies, and the many ways to portray similar ideas. The use of quantities in text, however, tends to be more reserved. The numerical relations developed by those quantities give insights into the concepts that a text is discussing, and additionally can act as a normalization of words across documents. For instance if one document mentions the phrase "10 feet above sea level" while another contains the words "6 meters tall", then a numerical relation extractor can find the common implicit relation of "height" and include this explicitly in the topic extraction model, thereby allowing algorithms to pick up on the fact that these two documents have something in common despite neither of the phrases sharing any common words. We explore the performance of text mining algorithms on documents modeled as the sum of their numerical relations.

Given a corpus of documents, we express each document as the sum of its numerical relation extraction tuples. We then create a document-word matrix of this reduced corpus and apply our topic modeling algorithms to this. We choose LDA and NMF for this as they are the two representative works of the two distinct approaches to topic learning [Chen et al., 2019]. As a first foray into this domain, we want to see how the most popular algorithms in the domain preform on our reduced data sets. Figure 3.2 shows this pipeline.

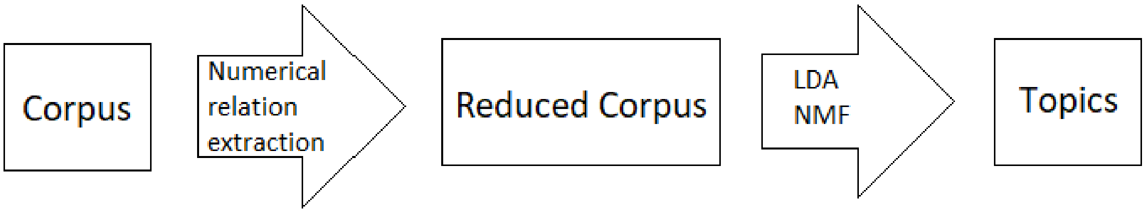


Figure 3.2: Topic modeling pipeline.

3.2 Patterns and Rules

From studying the dependency and POS parses of sentences, we come up with a number of patterns, the first of which is that an entity or a relation can consist of more than one noun phrase. We therefore call the set containing all noun phrases that reference the entity, the entity component and similarly for the relation. To generalize: a component is a chain of one or more connected noun phrases. One noun phrase is considered connected to the next unless one of the following tokens is present between the phrases: a verb, a conjunction, a punctuation, or, in the case of the entity, the closest determiner to the verb, the farthest proper noun preceding the verb, or the last possessive noun in an otherwise unbroken noun phrase chain. We then come up with the following rules for identifying which component is the **entity component** and which is the *relation component*, depending on their positions in the sentence relative to the verb and quantity: If the quantity is the noun subject of the verb:

- The entity component immediately follows the quantity and therefore precedes the verb. Any phrase after the verb is part of the relation. For example "10% of **the iceberg** is *above sea level*".
- The quantity is immediately followed by the verb, which is then followed by the relation component, and lastly, the entity component. For example "200m is *the height* of **the tower**".
- The quantity is immediately followed by the verb, which is then followed by the entity component which ends in a possessive noun that is then followed by the relation component. For example "7m is **the Great Wall of China**'s *height*".

If the quantity is not the noun subject of the verb, then one of the following observations hold. Note the first two observations of the following list allow for the quantity being in a different clause to the verb:

- The entity component immediately precedes the verb, the relation component precedes the quantity. For example "**Germany** has *a GDP* of \$3.7 trillion".
- The entity component immediately precedes the verb, the quantity is followed by a relation defining adjective. For example "**The tree** is 5 meters *tall*".
- The entity component ending in a possessive noun is followed by the relation component which is then followed by the verb and the quantity. For example "**France**'s *unemployment rate* is 8.5%"

It is worth noting that the most important component to identify is the entity component, as the relation component can be identified by looking for phrases either before or after the entity and the quantity. Using the above observations and the relevant POS and dependency tags, we can identify the entity, relation and quantity components, as well as the verb. All of the concepts introduced in this section are made formal in the following sections.

3.3 Preprocessing

Our model takes a text, D, as input. The input text is tokenized, annotated with POS tags, and parsed for inter-word dependencies. Co-reference resolution is then attempted, before splitting the text into its component sentences, such that:

$$D = \{S_1, \dots, S_N | N \text{ is \# of sentences in } D\}. \tag{3.1}$$

Every sentence $S \in D$ is a set of tokens $S = \{t_i\}_{i \in \{1,2,\dots,|S|\}}$, with every token $t \in S$ carrying its own POS tag, dependency tag, and position in S as attributes accessible by t_{pos}, t_{dep} and t_{ind} respectively. Each sentence is then parsed using a noun chunker to form S_{NP} which is the set of all noun phrases in S. Then

$$D_{num} = \{S \in D | \exists t \in S \text{ s.t. } t_{pos} = NUM\} \tag{3.2}$$

represents the set of sentences containing numerical values from which we extract relations. Any sentence in the complement set, $D \setminus D_{num}$, does not contain quantities, and is therefore removed.
Let $t^{parent} \in S$ denote the syntactic parent of token $t \in S$. Then:

- $t^{ancestors} = \{t^{parent}, (t^{parent})^{parent}, \dots, root\} \subset S$ denotes the set of all tokens in the syntactic ancestry of t until the syntactic root of the sentence.

- $t_{children} = \{t' \in S | t'^{parent} = t\}$ denotes the set of all syntactic children of t.
- $t_{descendants} = \{t' \in S | t \in t'^{ancestors}\}$ denotes the set of all syntactic descendants of t.

Let $q_S \in \{t \in S | t_{pos} = NUM\}$ be the token for a numerical value in sentence S. Define $v_q \in \{t \in q_S^{ancestors} | t_{pos} = VERB\}$ such that:

$$\{t' \in S | t'^{parent} = v_q \land t'_{dep} \in \{nsubj, nsubjpass\}\} \neq \emptyset$$
$$\text{and}$$
$$\forall y \in \{t \in q_S^{ancestors} | t_{pos} = VERB\}, y \in v_q^{ancestors}. \quad (3.3)$$

In other words, the token v_q is the most closely related syntactic ancestor verb of q_S that has a non-empty syntactic children set containing a noun subject. Then we can build a representation of the sentence that removes unnecessary tokens and problematic comma-separated clauses that are uninformative to the numerical relation, but may affect the extraction process. This representation is generated in the following steps:
Let S be a sentence and $q, v \in S$ be an instance of q_S and v_q defined as described above:

$$\begin{aligned} L1 = \{x \in S | (x \in \{q, v\}) \lor (x \in v_{descendants} \land x_{dep} \in \{poss, pobj, \\ dobj, attr, nsubj, nsubjpass, punct, det, cc\})\} \\ = \{x_k\}_{k \in \{1, \ldots, |L1|\}} \end{aligned} \quad (3.4)$$

The set $L1$ contains the noun, punctuation, determiner and conjunction tokens found in the clauses for which v is the syntactic root. If v is the syntactic root of the entire sentence S, then this will contain certain tokens from every clause; otherwise, this representation removes any clauses in S that are entirely independent of v.

$$\begin{aligned} L2 = \{y \in L1 | (y \in v_{children} \land y_{dep} \in \{nsubj, nsubjpass\}) \\ \lor (y_{dep} = punct) \lor (y \in \{q, v\})\} = \{y_j\}_{j \in \{1, \ldots, |L2|\}} \end{aligned} \quad (3.5)$$

We assume that only the comma-separated clauses containing v, q, and the noun subject of v are immediately relevant to the numerical relation. This assumption is necessary in order to simplify the extraction procedure, and for many of the cases we tested, it is also entirely appropriate. Note that q and the noun subject of v may be one and the same (i.e. $q_{dep} = nsubj$). The set $L2$ therefore contains these two or three word tokens alongside the punctuation tokens from $L1$.
Define mapping $\tau : \{1, \ldots, |L2|\} \to \{1, \ldots, |L1|\}$ such that $\tau(j) = k$ where $y_j = x_{\tau(j)} =$

x_k for $y_j \in L2, x_k \in L1$. τ is an injective function that maps the indices of tokens in $L2$ to their indices in $L1$. This is needed for the final two steps, in which the indices of consecutive punctuations in $L2$ are used to remove tokens found between said punctuations in $L1$, thereby removing entire clauses from the representation.

$$L2_{cpunct} = \{\tau(j) \in \{1, \ldots, |L1|\} | y_j, y_{j+1} \in L2 \wedge (y_j)_{dep} = (y_{j+1})_{dep} = punct\} \tag{3.6}$$

$$S_X = \{x_k \in L1 | \forall j \in L2_{cpunct}, k < \tau(j) \vee k > \tau(j+1)\} \tag{3.7}$$

Thus we arrive at our desired representation, S_X, of the sentence S containing the tokens that we require to build the extraction. Note, however, that S_X only holds for a particular q and v pair; each quantity q_S in the sentence may be syntactically related to a different verb v_q and thus produce a different S_X. This concludes the preprocessing step.

3.4 Quantity Extraction

The extraction steps attempt to define $S_{out} = (E, R, Q)$, a triple of an entity component E, numerical quantity component Q and the relation component, R, for every q_S in every S in the document D. In the quantity extraction step, we define the set Q. In this and the following sections we focus on the extraction of a single instance q with verb v in a sentence S; this process can then be extrapolated to every such q_S, v_q and S in D. We parse the tokens surrounding q in S using a quantity extractor that uses regular expressions to locate contiguous number and measurement unit combinations that form quantities. The number q, if spelled out, is converted to numeric digits, while its unit token(s) q_{unit}, if abbreviated, is written out. If the number is a range, then the quantity extractor returns the average of the two extremes. We define the set $Q = \{q, q_{unit}\}$. If the quantity extractor fails to recognize the unit of the quantity, we instead define $Q = \{p \in S_{NP} | q \in p\}$ to be the noun phrase containing both the value and unit of the quantity.

The quantity extractor uses an internal knowledge base of recognized measurement units and their measures to look up the measurement unit q_{unit} and identify its corresponding measure m (e.g. meter - length, \$ - currency). This measure, if it exists, helps form the basis of the relation component R of the extraction, and is therefore retained for the relation extraction step. In cases where q_{unit} does not correspond to any measure, m is undefined.

3.5 Entity extraction

The next step is to define the set E containing the token(s) referencing the entity of the relation. We first need to find a token e in S_X that is part of the entity set E; note that e itself is not necessarily the entity, but a token in the noun phrase that makes up the entity. There are two cases we must consider: either the quantity is the noun subject of the relation, or it is not. Within each case, there are a number of possibilities for e; the first instance, in the order below, for which an e can be defined is accepted.

3.5.1 Quantity is the Noun Subject

- The first possibility is that the entity occurs between the quantity q and the verb v as in the sentence "2% of **Germany** is uninhabited". Define $e = \min_{x_{ind}}(\{x \in S_X | x_{pos} \in \{DET, PROPN, NOUN\} \wedge q_{ind} < x_{ind} < v_{ind}\})$ as the first noun token or determiner (which heralds the beginning of the entity phrase) token after q.

- If such an e does not exist, the next option is that the entity occurs after the verb and is a possessive noun that precedes a relation word, as in the sentence "260,000 km2 is **Germany**'s area". In this case, define $e = \min_{x_{ind}}(\{x \in S_X | x_{dep} = poss \wedge x_{ind} > v_{ind}\})$.

- If such a possessive noun e does not exist, then we consider the next possibility. When a relation word immediately follows the verb, it is preceded by a determiner. The entity then follows the relation word and may also be preceded by a determiner, as in the sentence "200 m is the height of **the tower**". In this case we look for the second determiner token following the verb, and thus define $e = min_{x_{ind}}(\{x \in S_X | x_{dep} = det \wedge x_{ind} > v_{ind}\} \setminus \min_{x_{ind}}(\{x \in S_X | x_{dep} = det \wedge x_{ind} > v_{ind}\}))$.

- If such a token e does not exist, we look for the first proper noun following the verb, as in the sentence "260,000 km2 is the area of **Germany**". Define $e = \min_{x_{ind}}(\{x \in S_X | x_{pos} = PROPN \wedge x_{dep} = pobj \wedge x_{ind} > v_{ind}\})$.

- If such an e does not exist, we look for the first noun after v that is the object of a preposition as opposed to the direct object. Define $e = \min_{x_{ind}}(\{x \in S_X | x_{dep} = pobj \wedge x_{ind} > v_{ind}\})$. This captures both the example sentences in the previous two possibilities, however will not return the correct e if the relation words between v and e also contain nouns that are the object of a preposition as the word "capita" is in the phrase "GDP per capita".

3.5.2 Quantity is not the Noun subject

If q is not noun subject, then there is considerably less complexity. The entity is usually the noun phrase immediately prior to the verb, except if the entity is a possessive noun.

- If the entity is a possessive noun, then define e as the last possessive noun prior to the verb. Define $e = \max_{x_{ind}}(\{x \in S_X | x_{dep} = poss \wedge x_{ind} < v_{ind}\})$. This accounts for sentences such as "**Germany**'s area is 260,000 km2".
- If no such possessive noun exists then we assume the entity phrase includes the last noun before the verb, as in the sentence "the height of **the tower** is 200m". In this instance, define $e = \max_{x_{ind}}(\{x \in S_X | x_{ind} < v_{ind}\})$.

Once e has been defined, we finally define the entity set $E = \{p \in S_{NP} | e \in p\}$ as the phrase containing the token e.

3.6 Relation Extraction

With E and Q now defined, all that is left is to define the relation set R. We first create a set R' that consists only of the relation words explicitly stated in the sentence S, and then by combining R' with an implicit relation name r that we find using the measure m (described in the quantity extraction section), we can define R.

If the quantity is the noun subject, and the entity token e occurs before the verb v, all the tokens following the verb in that clause are part of the relation. i.e. if $q_{ind} < e_{ind} < v_{ind}, R' = \{x \in S_X | x_{ind} > v_{ind}\}$. An example of this is the sentence "2% of Germany is **uninhabited**". For all other cases, we define $R' = \{x \in S_X | x \notin E \cup Q \wedge x_{dep} \in \{pobj, dobj, attr, nsubj, nsubjpass\}\}$, which is the set of all nouns and adjectives (if $x_{dep} = attr$ and $x_{pos} = ADJ$) in the clause(s) represented by S_X that are not part of the quantity or entity sets.

To find implicit relation name r, we make use of a second knowledge base that links measures, m, to possible relation names, r, via their common explicit keywords (see Table 3.1). If R' contains any of the keywords listed in the knowledge base under m, we assign the associated relation name to r. We construct this knowledge base manually, handpicking the keywords and relation names, adding or removing them depending on how many distinctions between similar relations we wish to make. For instance, we can use this knowledge base to distinguish between relations sharing the same units (e.g. width versus height) by detecting the presence of one or more of their associated keywords (e.g. wide vs tall) in the set R'. If such a distinction is not necessary, then we can combine the keywords for these relations and link them with a common relation (e.g. length), thus simplifying the relation. If no keywords relating m to r are found in R', set $r = m$ and proceed. If m is not defined, then r is simply ignored. Define $R = \{r\} \cup \{p \in S_{NP} | x \in R' \cap p\} \cup \{x \in R' | x_{pos} = ADJ\}$ as the set containing the implicit relation name r and the relation words in the form of noun phrases and adjectives. Having defined Q, E and R, we output $S_{Out} = (E, R, Q)$. This entire process is then repeated for each quantity q in each sentence S in document D. We define $d = \{S^q_{out} | \forall q \in S, \forall S \in D_{num}\}$.

Measures (m)	Explicit Keywords	Implicit Relation (r)
length	long, longest	length
	height, high, tall	height
	width,wide	width
currency	gdp, gross domestic product	gdp
percentage	interest, interest rate	interest rate

Table 3.1: Optional KB to identify implicit relation r given a sentence containing at least one of the explicit keywords under a measure m.

3.7 Topic Modeling

We now describe the second part of our model that explores the usefulness of numerical relations as a means of dimension reduction in topic modeling. Let $\mathcal{D}$ be the corpus consisting of M documents D. Using the relation extraction procedure described above we can define $\mathcal{D}_{extractions} = \{d_1, d_2, \ldots, d_M\}$ as the set containing the extractions found in each document. Note that we lemmatize the words in the extractions for the topic mining task. We use the LDA and NMF algorithms introduced in Section 2.4 on the reduced corpus $\mathcal{D}_{extractions}$

3.7.1 LDA

We model the generation of documents using LDA as follows:

1. For each document $d \in \mathcal{D}_{extractions}$, sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$ where $\text{Dir}(\alpha)$ is the Dirichlet distribution parameterized by α. α is the prior of the document-topic distribution.

2. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$. β is the prior of the topic-word distribution.

3. For each word w_n in document d where $n \in \{1, 2, \ldots, |d|\}$:

a) sample a topic $z_i \sim \text{Mult}(\theta_d)$ where $i \in \{1, 2, \dots, K\}$ and $\text{Mult}(\theta_d)$ is the multinomial distribution

b) sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

Note that w_n represents the words in the document, rather than the tokens, the distinction being that a token has a POS and dependency tag, whereas a word does not. Given the parameters α, β, and K (which we initialize in the next chapter when we describe our data), the joint distribution of the model is:

$$p(\phi_{1:K}, \theta_{1:M}, z_{1:K}, w_{1:|d|}|\alpha, \beta) = \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{d=1}^{M} p(\theta_d|\alpha) \prod_{n=1}^{|d|} p(z_{d,n}|\theta_d) p(w_{d,n}|z_{d,n}) \tag{3.8}$$

Given the observed word distribution in each document, the posterior distribution of the hidden variables can be calculated as:

$$p(\phi, \theta, z|w, \alpha, \beta) = \frac{p(\phi, \theta, z|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{3.9}$$

This posterior is intractable to compute. In this thesis, we use an implementation proposed by [Hoffman et al., 2012] that uses a variational Bayesian method with a simpler distribution $q(\phi, \theta, z|\lambda, \psi, \gamma)$ to approximate it, where the parameters λ, ψ, γ are optimized to maximize the Evidence Lower Bound L:

$$\log\ p(w|\alpha, \beta) \geq \mathbb{E}_q[\log\ p(w, z, \theta, \phi|\alpha, \beta)] - \mathbb{E}_q[\log\ q(z, \theta, \phi)] \triangleq L(w, \psi, \gamma, \lambda) \tag{3.10}$$

which is the equivalent of minimizing the divergence between the approximation distribution and the true posterior distribution. The resulting distribution $q(\phi, \theta, z|\lambda, \psi, \gamma)$ gives us the topic-word and document-topic distributions as required [Blei et al., 2003, Jónsson and Stolee, 2015, Allahyari et al., 2017].

3.7.2 NMF

We define the vocabulary $V_{extractions}$ as the set of all unique words across the documents in $\mathcal{D}_{extractions}$. Let N be the number of unique words in our reduced corpus, i.e. $|V_{extractions}| = N$. Then we create the $M \times N$ document-word matrix A of $\mathcal{D}_{extractions}$. This matrix is non-negative making it a prime target for NMF, which we use to find two lower dimensional non-negative matrices W and H such that

$$A \approx WH \tag{3.11}$$

where $W \in \mathbb{R}_{+}^{M \times K}, H \in \mathbb{R}_{+}^{K \times N}$. The matrix A is the document-term matrix of the corpus; W is the matrix consisting of K basis vectors of A, which represents the K clusters discovered in the vocabulary of the corpus, while H is the coefficient matrix that represents the weight of each cluster in each document [Lee and Seung, 1999]. We provide the value of K in the next chapter after describing our data. We calculate W and H by optimizing over the squared Frobenius norm:

$$\frac{1}{2}||A - WH||_{Fro}^{2} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(A_{ij} - (WH)_{ij})^2 \tag{3.12}$$

To provide initial values for W and H we use the NNDSVDa method proposed in [Boutsidis and Gallopoulos, 2008] that creates initial guesses for W and H by processing together the three matrices produced by the singular value decomposition of matrix A. One of the most popular ways to optimize the above equation uses the multiplicative update scheme as proposed by [Lee and Seung, 2001] where W and H are iteratively improved by the following rules until convergence:

$$W_{i,k}^{New} \leftarrow W_{i,k}\frac{(AH^T)_{ik}}{(WHH^T)_{ik}} \tag{3.13}$$

$$H_{k,j}^{New} \leftarrow H_{k,j}\frac{(W^TA)_{kj}}{(W^TWH)_{kj}} \tag{3.14}$$

Upon convergence, W and H will give us the topic-word and document-topic matrices respectively.

3.8 Summary

In this chapter we defined our rule-based attempt at numerical Open IE, which decomposes a sentence into its quantity-relevant clauses and then uses grammatical rules to identify the entity, quantity and relation components based on the position of the quantity relative to its syntactic parent verb. Additionally, we presented our idea of reducing a corpus down to its numerical relations based on the idea that relation extractions are informative and can introduce implicit concepts that can be used to identify topics using LDA and NMF. In Chapter 4 we evaluate the performance of our numerical relation extractor versus its counterparts, and we compare the topics identified in our reduced corpus with the topics identified in an unreduced corpus.

4 Experimental Evaluation

In this chapter, we define the evaluation metric, test data and the parameter settings used for each task. Section 4.1 describes our model implementation. Section 4.2 describes the experimental setup for numerical relation extraction and then presents and discusses our results, while Section 4.3 does the same for topic modeling.

4.1 Model Implementation

Unlike the previous works in numerical relation extraction, we choose to implement our entire model in Python 3.6. We make use of the spaCy[1] library alongside its NeuralCoref[2] pipeline extension for our text processing and co-reference resolution requirements. The quantity extractor and unit-measure KB we use in the model is from the quantulum3[3] library. Lastly, our model uses the scikit-learn [Pedregosa et al., 2011] implementations of the LDA and NMF algorithms for topic modeling.

4.2 Evaluating Relations Extractions

Evaluating the quality of an extracted relation tuple is no easy task. Despite many Open IE models being proposed over the last thirteen years since TEXTRUNNER [Etzioni et al., 2008], there still has not been a well defined specification for was constitutes a valid extraction. The previous Open IE systems have largely been evaluated by hand on small corpus consisting of a few hundred sentences. This naturally goes against one of the purposes of Open IE, which is its ability to scale to large amounts of text. Additionally, evaluation by hand is an arduous task that is open to differing perspectives of what constitutes a good extraction. Such systems often use precision or yield (the number of correct extractions as deemed by the evaluator) as metrics for the efficacy of a model [Niklaus et al., 2018]. [Saha et al., 2017] use these two metrics in their evaluation

[1] https://github.com/explosion/spaCy
[2] https://github.com/huggingface/neuralcoref
[3] https://github.com/nielstron/quantulum3

of BONIE, the numerical Open IE system. A work by [Stanovsky and Dagan, 2016] in 2016 proposed a set of guiding principles for the evaluation of Open IE systems:

- Given the sentence “Sam succeeded in convincing John”, most Open IE systems output the tuple *(Sam, succeeded in convincing, John)* rather than the implied relation of *(Sam, convinced, John).* This is generally agreed upon approach across the Open IE models and is known as the **Assertedness** principle: Extracted relations should be asserted by the original sentence.

- Relations should be kept as minimal as possible as long as the original information is preserved. This means that one of the entities may need to be split. For example, the sentence “Bell distributes electronic and building products” should form two extractions: *(Bell, distributes, electronic products)* and *(Bell, distributes, building products).* This is the **Minimal propositions** principle.

- Lastly, the **Completeness and open lexicon** principle states that Open IE systems should aim to extract all asserted relations in a sentence, not only a limited scope.

Since the field of numerical relation extraction is still developing, we are unable to find any explicit guidelines. Therefore we follow the lead of the BONIE model and use yield as our metric; however, we will compare our extractions to BONIE’s extractions according to the three principles stated above.

4.2.1 Test Data

Since there is only one existing numerical Open IE system, there is a dearth of labeled text rich in quantities. Generating such data is extremely resource intensive; for this reason, we have decided to use the same test data set [4] that BONIE does. This text data set consists of 2000 numerical sentences that are randomly selected from ClueWeb12 [5]. The sentences consist of all manner of natural language, and many also suffer from grammatical inaccuracies and also semantic incomprehensibility. However, since we are comparing our model to the BONIE model, which also uses grammar based parsing of sentences to develop its pattern learning, such sentences pose an a common issue.

4.2.2 Parameter Setting

Our model takes in an optional KB for narrowing down identified measures. Since it is unfeasible to manually identify all the possible measures in a test set of 2000 sentences,

[4] https://github.com/dair-iitd/OpenIE-standalone/tree/master/data

[5] http://www.lemurproject.org/clueweb12.php/

we restrict ourselves to only tuning a few of the measures as seen in Table 4.1.

Measures (m)	Explicit Keywords	Implicit Relation (r)
length	length, long, longest	length
	height, high, highest, elevation, above sea level, altitude, tall	height
	width, wide, breadth	width
currency	gdp, gross domestic product	gdp
	stock price	stock price
percentage	interest, interest rate	interest rate
	unemployment rate	unemployment rate

Table 4.1: Relation tuning used on test data set.

4.2.3 Relation Extraction Results

Model	Yield
NumberRule	6
BONIE (seed patterns only)	72
BONIE	458
Our Model	111

Table 4.2: Yield (number of correct numerical extractions) on the test data set of 2000 ClueWeb12 numerical sentences.

The resulting yields of our rule-based model and the BONIE model are shown in Table 4.2. We also include the results, as reported by [Saha et al., 2017], of the only other rule-based numerical relation extractor, NumberRule, in the table. The NumberRule model performed particularly poorly on our test data. This is likely due to the fact that it requires a predetermined list of relations which is difficult to do for free text, and because it uses Named Entity Recognition to identify the entity. The test data set has very few sentences containing known entities, making the current iteration of NER hopeless for finding the entity. Our model is able to extract the numerical relations from 111 sentences completely, which is more than the bare-bones, unlearning, BONIE model that only uses its seed patterns, but far fewer than the full learning BONIE model. However, of the sentences that we extract correctly, BONIE only outputs extractions for 68 of them, meaning our model correctly extracts 42 sentences that BONIE cannot. Additionally, our model is able to extract a further 37 relations correctly from sentences containing more than one relation, bringing the total number of corrected extracted relations to 148. BONIE on the other hand, does not extract more than one relation from each sentence, and therefore breaks the **minimal propositions** principle mentioned in Section 4.2. Table 4.3 shows some examples of this. Another observation is that the BONIE model does not deal as well as our model with noun phrase relations (as opposed to verb mediated relations) in cases where the relation precedes the entity. The **completeness and open lexicon** principle therefore suggests that our extractions are better in the examples also shown in Table 4.3.

Minimal Propositions		
Sentence	This recipe yields 6 supper servings, 12 appetizer servings.	As a Birthday gift, she purchased 2 Cohiba and 2 Partagas Maduro's.
BONIE extractions	(This recipe, yields, 6 supper servings)	(she, purchased, 2 cohiba, As a Birthday gift)
Our models extractions	(This recipe, yields, 6 supper servings)	(she, purchased, 2 Cohiba)
	(This recipe, yields, 12 appetizer servings)	(she, purchased, 2 Partagas Maduro)
Completeness and Open Lexicon		
Sentence	In 1996, Israel's GDP per capita was $17,200.	Maximum price per ticket is $100.
BONIE extractions	(Israel's GDP per capita, was, $ 17200, In 1996)	(Maximum price per ticket, is, $ 100)
Our models extractions	(Israel, GDP per capita, is, 17200.0 dollar)	(ticket, Maximum price, is, 100.0 dollar)

Table 4.3: Examples of test sentences where our model output a better extraction than BONIE under the minimal propositions principle and the completeness and open lexicon principle.

Unfortunately our model also has several problems. For 205 instances in the corpus, our model identified a punctuation as the entity. Generally, our model had difficulty in selecting the correct entity and often repeated phrases in the relation component that were already in the entity or quantity components. This is down to ill-specified boundaries and limitations in our set of rules on what constitutes an entity, as well as our reliance on co-reference resolution, which is a field that, while showing signs of improvement, still has some ways to go. This is also the reason why a metric such as *precision* would have given a very poor result since our model attempted to put out relations for almost all the test sentences. Additionally, because we did not optimally narrow down certain measures in the optional KB we provided, a few measures created unnecessarily wordy relation phrases such as “mass of total metal weight” when simply “total metal weight” would have sufficed. Such extractions therefore violate the **minimal propositions** principle. Another issue was that we output the verb phrase of the relation separate to the noun phrase of the relation, thereby occasionally creating awkward extractions where the verb follows the noun relation in cases where it should precede it. Since we relied on a quantity extractor to identify the quantity in the sentence, we missed out on quantity modifiers such as the word “over” in the phrase “over 10 percent”. Due to the nature of numerical relation extraction, both BONIE and our model occasionally violated the **assertiveness** principle whenever implicit relation words where included. Overall, our model gave a higher yield than other non-learning models, and of the correctly extracted sentences, sometimes gave briefer and better quality extractions than BONIE, sometimes equivalent extractions, and sometimes worse (technically correct, but awkwardly phrased). BONIE, however, was still able to output more than quadruple the number of our extractions.

4.3 Evaluating Topic Models

Unlike for relation extraction, there are several established methods of evaluating topic models such as *pointwise mutual information* as proposed by [Newman et al., 2010] that measures the coherence of a topic z_i as:

$$Coh_i = \frac{2}{K(K-1)} \sum_{j<k\leq K} \log \frac{p(w_j, w_k)}{p(w_j)p(w_k)} \tag{4.1}$$

where K is the number of most probable words in each topic, and $p(w)$ is the probability of word w appearing in a document. Unfortunately, this method does not perform well on short texts (which our documents become post-dimension-reduction) because topic coherence is measured by word co-occurrence which can be low in smaller text sizes [Quan et al., 2015, Qiang et al., 2019]. Another proposal is *purity* [Manning et al., 2008], which is a clustering evaluation technique. To calculate purity, we choose the maximum topic probability for each document, and assign the document to that topic. Each topic can therefore be represented as a set of the documents assigned to it. The contents of these sets are then compared against the actual classifications of the main topic of each document as provided by us through manual labeling. Formally:

$$purity(Z, \mathcal{C}) = \frac{1}{M} \sum_{i=1}^{K} \max_j |z_i \cap c_j| \tag{4.2}$$

where $Z = \{z_1, z_2, \ldots, z_K\}$ is the set of K topics, and $\mathcal{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of the actual class of the documents. M is the number of documents in the test corpus. One of the issues with the purity metric is that if each document gets assigned to a distinct topic (i.e. there are no two documents with the same topic) then the above equation returns a purity of 1, which is extremely unhelpful. However, we do not run this risk since we ensure the actual number of topics that is in the test data and it is far less than the number of documents. Therefore, we will use purity as our evaluation metric on the results of the LDA and NMF algorithms. We run the following experiments for comparison:

- Original text in the corpus.
- Only extraction tuples as provided by running our numerical relation extractor on the corpus.

- Only the relation components and the units of the quantities in the extraction tuples.
- Manually labeled numerical relation tuples (the "correct" extraction) in the corpus.
- Only the relation component and the unit of the quantity of the manually labeled numerical relations.

For each of the experiments above, we filter out the stop words, punctuation and the numerical values of the quantities across the corpus. The removal of punctuation and stop words is unremarkable, but our decision to remove the values is based on the fact that the topic modeling algorithms attempt to find co-occurring values regardless of the fact that the values are not in any way related to one another.

4.3.1 Test Data

Again we face the problem of a lack of existing labeled quantity data. We cannot use the data set from the relation extraction task because it does not contain topic labeling. For this task we therefore build our own corpus consisting of 33 short documents generated from Wikipedia text. The articles selected for this task broadly fall into four classes:

- *Astronomical bodies* consisting of descriptions of the physical characteristics of 10 bodies in our solar system including quantities such as diameter, mass and orbit duration.
- *Elements* consisting of descriptions of the attributes of 10 elements from the periodic table, including quantities such as atomic number, atomic weight and periodic group. A number of these elements also occur in some of the descriptions of astronomical bodies, thus creating an overlap.
- *Landmasses* consisting of descriptions of the physical characteristics such as area, population and surrounding bodies of water of 8 landmasses (5 continents and 3 countries).
- *Water bodies* consisting of descriptions of the physical characteristics of 5 bodies of water (3 oceans and 2 seas) including information such as surface area, depth and surrounding landmasses. This again creates topic overlap in some of the documents.

The sentences in the corpus are only filtered to remove dates, all other features are kept intact. The corpus contains multiple sentences with no numerical relations, but every document contains at least one sentence with a quantity. Some documents contain a degree of overlap, which may or may not occur in the numerical relation. We allow for this to happen because it is a feature of free text that quantities are used less frequently and therefore may only capture certain topics in the data.

4.3.2 Parameter Settings

Our test corpus consists of four classes, therefore we attempt to find 4 topics in the corpus using LDA and NMF (i.e. $K = 4$). The LDA algorithm also requires values for the document-topic prior, α, and the topic-word prior, β. We set $\alpha = \frac{1}{2}$ since most of the documents contain two topics, and we set β to the default value of $\beta = \frac{1}{K} = \frac{1}{4}$ since we have no information to the contrary.

4.3.3 Topic Modeling Results

The results of the LDA and NMF algorithm runs on our five versions of the Wikipedia corpus are shown in Table 4.4, which shows the topic assignment for each document where the probability of said topic is over 0.3. We set this condition so that we can see which documents are harder to classify. As we can see from the table, both algorithms had a difficult time differentiating between landmasses and bodies of water based on the original text. Overall, NMF was able to perform better than LDA on the original text as it was able to correctly distinguish the astronomical bodies form the elements. Our extractions provided a better split of the land and water classes, but still contained several misclassifications. However, this time LDA was better able to split the space entities from the elements. NMF actually did not identify the elements as a topic at all, rather the lack of one (see Table 4.5). The 3rd experiment using the relations and units component of our extractions served only to worsen the results of both extractions, with LDA grouping elements and astronomical bodies into one topic, and NMF still being unable to form a topic specific to the elements. The LDA algorithm performed significantly better on the corpus of manually annotated tuples, with only 3 misclassifications. NMF posted similar results as for the extracted tuples run, still unable to create a topic specifically for elements (see Table 4.6). The final corpus containing only the relation-relevant components of the manually annotated extractions generated the best results for NMF, which was finally able to create a topic (see Table 4.7) for elements and managed to classify 30 of the 33 documents. The performance of LDA, however, declined as it became unable to distinguish between elements and landmasses.

Both algorithms faced significant problems differentiating between landmass and body of water on the original text because in both cases the document would often mention

		Original		Extract		Extract (Rel)		Labels		Labels (Rel)	
Class	Subject	LDA	NMF	LDA	NMF	LDA	NMF	LDA	NMF	LDA	NMF
Astronomical body	Sun	0	0	0	2	1	3	2	1	3	1
Astronomical body	Jupiter	0	0	0		1	2	2	1	3	1
Astronomical body	Earth	0	0	0	2	1	2	2	1	3	1
Astronomical body	Mercury	0	0	0	2	1	2	2	1	3	1
Astronomical body	Venus	0	0	0	2	1	2	2	1	3	1
Astronomical body	Mars	0	0	0	1	0	1	2	2	3	1
Astronomical body	Saturn	0	0	0	2	1	2	2	1	3	1
Astronomical body	Neptune	0	0	0	2	1	2	2	1	3	1
Astronomical body	Uranus	0	0	0	2	0	2	0	1	3	1
Astronomical body	Pluto	0	0	0	2	1	2	2	1	3	
Element	Carbon	3	3	3	2	3	3	3		2	3
Element	Chlorine	3	3	3		1		3		0	3
Element	Helium	2	3	3		1		3		0	3
Element	Hydrogen	2	3	0		1	3	3		0	3
Element	Iron	2	3	3		1		3		0	3
Element	Oxygen	2	3	3		1		3		0	3
Element	Sodium	2	3	3		1		3		0	3
Element	Gold	3	3	3		1		3		0	3
Element	Copper	3	3	3		1		3		0	3
Element	Mercury	2	3	3		1		3		0	3
Landmass	US	1	1	2	0	0	0	1	3	0	0
Landmass	North America	1	1	1	1	0	1	1	2	0	0
Landmass	Europe	1	1	2	1	0	1	1	2	0	0
Landmass	Asia	1	1	2	1	0	1	1	2	0	0
Landmass	Africa	1	1	1	1	0	1	0	2	0	0
Landmass	Russia	1	1	1	1	3	2	0	2	0	0
Landmass	South America	1	1	1	1	2	1	1	3	0	0
Landmass	Germany	0	2	2	0	3	0	1	2	0	0
Water body	Atlantic Ocean	1	1	1	3	2	1	0	0	2	2
Water body	Pacific Ocean	1	1	1	3	2	1	0	0	2	2
Water body	Indian Ocean	1	1	1	3	2	1	0	0	2	2
Water body	Mediterranean Sea	1	1	0	3	2	1	0	0	2	2
Water body	Caspian Sea	1	1	2		2	1	0		2	

Table 4.4: Topic assignment for the 5 versions of the test corpus of 33 documents for LDA and NMF algorithms. (Rel) in the column heading denotes that the corpus consists of only the relation component of the extraction tuple. A topic is only assigned to a document if the probability of the topic being in the document is greater than 0.3.

Numerical Relation Extraction Tuples of Corpus

LDA Keywords by Cluster								NMF Keywords by Cluster							
0		1		2		3		0		1		2		3	
sun	14.2	percentage	13.4	area	12.5	atomic	11.2	angle	2.4	percentage	1.9	sun	1.7	ocean	1.7
earth	12.5	area	11.0	angle	8.2	number	10.2	km2	1.5	population	1.7	earth	1.4	volume	1.5
planet	10.2	kilometre	9.0	europe	8.2	element	4.2	area	1.3	area	1.4	mass	1.4	percentage	1.4
time	9.3	ocean	8.2	square	7.5	14c	3.2	degree	1.2	america	1.2	time	1.4	metre	1.2
diameter	9.2	population	7.4	kilometre	7.5	half	2.3	land	0.9	europe	0.9	diameter	1.0	kilometre	1.2
mass	9.2	america	7.3	percentage	7.2	year	2.3	square	0.9	north	0.8	planet	0.9	depth	1.0
percentage	6.2	metre	7.2	km2	5.2	centavo	2.2	kilometre	0.9	world	0.8	year	0.7	indian	0.8
neptune	5.2	square	7.0	total	4.4	life	2.2	water	0.9	earth	0.7	neptune	0.6	square	0.7
average	4.9	volume	6.4	large	4.3	stable	2.2	germany	0.9	square	0.7	ton	0.5	area	0.6
kilometre	4.3	depth	6.2	land	4.3	radionuclide	2.2	europe	0.7	kilometre	0.6	mars	0.5	length	0.6

Table 4.5: Top 10 keywords by topic identified by LDA and NMF in the Extracted corpus.

Relation Component and Quantity Unit of Extraction Tuples of Corpus

LDA Keywords by Cluster								NMF Keywords by Cluster							
0		1		2		3		0		1		2		3	
percentage	15.9	atomic	11.2	kilometre	8.7	angle	8.2	angle	2.3	percentage	2.4	planet	1.6	time	1.6
area	11.4	planet	10.2	percentage	8.7	km2	5.2	km2	1.4	area	1.5	sun	1.0	mass	1.2
population	8.2	number	10.2	metre	7.2	area	4.9	area	1.3	kilometre	1.2	average	1.0	year	1.2
world	6.3	time	9.8	area	6.4	degree	4.2	degree	1.2	square	0.9	earth	0.9	ton	0.8
earth	6.3	sun	8.2	square	6.0	land	4.2	square	0.9	population	0.9	diameter	0.8	atomic	0.6
kilometre	5.5	earth	8.2	volume	4.2	kilometre	4.1	kilometre	0.9	world	0.7	eighth	0.6	number	0.5
square	5.5	diameter	6.2	depth	4.2	square	3.3	land	0.9	metre	0.6	distance	0.6	earth	0.4
fourth	5.1	mass	5.2	length	3.4	half	3.1	large	0.6	total	0.5	time	0.6	length	0.4
total	4.5	element	4.2	surface	3.2	large	2.3	consist	0.6	earth	0.5	fourth	0.4	life	0.4
country	3.3	average	4.2	average	2.3	consist	2.2	europe	0.6	surface	0.4	large	0.4	stable	0.4

Table 4.6: Top 10 keywords by topic identified by LDA and NMF in the relations component of the extracted corpus.

Relation Component and Quantity Unit of Labeled Tuples of Corpus

LDA Keywords by Cluster								NMF Keywords by Cluster							
0		1		2		3		0		1		2		3	
area	18.3	degree	2.2	depth	8.2	time	17.2	area	2.5	time	2.0	depth	1.3	atomic	1.8
population	13.2	water	2.2	meter	8.2	earth	13.4	population	1.6	mass	1.6	meter	1.3	number	1.6
atomic	11.2	land	1.8	area	8.2	mass	11.2	earth	1.3	earth	1.4	earth	1.2	group	0.8
km2	10.3	north	1.2	surface	7.3	sun	9.2	km2	1.2	sun	1.2	area	1.2	table	0.6
number	10.2	east	1.2	volume	7.2	diameter	8.2	land	0.8	diameter	1.0	water	1.1	periodic	0.6
earth	8.2	latitude	1.2	earth	6.2	year	4.6	state	0.7	year	1.0	surface	1.0	standard	0.2
state	5.2	longitude	1.2	water	5.3	kilometer	4.2	surface	0.6	second	0.6	volume	1.0	weight	0.2
land	4.7	km2	0.3	km2	5.2	orbit	4.2	world	0.6	ton	0.6	km2	0.6	universe	0.2
world	4.3	state	0.3	group	4.2	day	3.2	degree	0.3	rate	0.6	average	0.5	baryonic	0.2
surface	4.2	day	0.3	average	3.3	ton	2.2	territory	0.2	kilometer	0.5	percent	0.5	half	0.2

Table 4.7: Top 10 keywords by topic identified by LDA and NMF in the relations component of the manually labeled corpus.

the other class, for example “The Indian Ocean is bounded by Asia to the north, Africa to the west, and Australia to the east”. Such a sentence would not cause any problems for the extracted or manually labeled corpus, because it contains no numerical relations. The elements class caused difficulties for the extracted and manually labeled corpora because many of the element documents only contained a single relation. The NMF algorithm therefore largely ignored this relation and instead gave such documents no clear topic at all. However, if we remove the requirement of the topic probability being above 30%, thereby forcing each document to be assigned to a topic, then we get the purity values as shown in Table 4.8. The NMF algorithm now gives topics to the elements class for the extracted tuples, but still does not change the result that the best clustering is achieved by NMF on the final corpus consisting of only the manually labeled relations and units. The reason for this being better than its full tuple counterpart is that the entity name appears in every extraction tuple thus giving it a high topic weight; therefore if the entity is mentioned in a document in a different class, the high weight of the entity can swing the topic probability, thereby causing a misclassification. From the purity table we can also see that there is no clear pattern on which of the algorithms performs better. From this set of experiments we can conclude that using relations from a numerical relation extractor that is able to create high enough quality extractions may perhaps be a viable way of dimensionality reduction with the aim of document clustering. For topic modeling, however, the question remains open since there are clearly words in quantity-less sentences that affect the assigned topic as shown by the difficulty of clustering the elements into a single class for the extracted corpora.

	Original		Extract		Extract (Rel)		Labels		Labels (Rel)	
	LDA	NMF	LDA	NMF	LDA	NMF	LDA	NMF	LDA	NMF
Purity	0.82	0.85	0.82	0.67	0.64	0.76	0.91	0.67	0.73	1.00

Table 4.8: Purity table

4.4 Summary

In this chapter we described the test data sets and the evaluation metrics we used for each of the two tasks. We then presented the results of our experiments and found that our numerical relation extractor shows mixed results - it has a high yield compared to its rule-based counterpart, but a much smaller yield than a learning system. The extractions themselves, if correct, can occasionally be awkwardly phrased, but in several cases follow the principles described by [Stanovsky and Dagan, 2016] better than the BONIE model. From the results of the topic modeling task we learned that using numerical relations as a means of dimension reduction may work for document clustering better than it does for topic modeling, but only if sufficiently high quality numerical relation extractions are used, which, given the relative inactivity of the field of numerical relation extraction, may not be possible yet.

5 Conclusions

In this thesis, we set out to create the first rule-based open information extraction system specifically designed for numerical relations. Our model, inspired by insights from previous works in both numerical and non-numerical relation extraction, was based on the generalization of syntactic patterns found in easier sentences to more complex sentences. We used the discovered patterns to define rules for the positioning of entity, relation, and quantity components around a relation-defining verb. We built a preprocessing pipeline capable of removing comma-separated, relation-irrelevant subclauses from sentences, and by our rules on the resulting filtered sentence representation, we were able to extract tuples of the form *(entity, relation, quantity)*. By studying the units of the quantity phrase, we were, in certain cases, able to imply measures even if they were not explicitly part of the sentence, and include these in the extracted output. We compared our results to BONIE, the only other Open IE model in the field of numerical relation extraction. As an extension of our first goal, we studied the efficacy of numerical relations as a means of dimension reduction in topic modeling. We tested the performance of two of the most popular algorithms for topic modeling: Latent Dirichlet Allocation and Non-negative Matrix Factorization. We evaluated the performance of these algorithms on the same corpus under five different dimension reducing filters.

5.1 Discussion

In the relation extraction task, we found that it is very difficult, if not impossible, to construct purely grammar-based rules capable of distinguishing whether a noun phrase is part of an entity or the relation. The results of NumberRule suggest that the other approach to solving this problem, Named Entity Recognition, is not yet capable of producing decent results on free text. Similarly, we found that the extractions generated by a syntax-focused model are limited by the part-of-speech tagger, dependency parser and co-reference resolution software used, and that even if they were to work perfectly, a large part of free text does not observe many grammatical rules and thus defies extraction by such means. We found that there is no established Open IE model evaluation technique, only guidelines for manual annotation. In terms of results, we found that our numerical relation extractor was able to output more correct extractions than the other non-learning models in the domain, but was inferior to the learning model in terms of yield. The correct extractions made by our model were occasionally awkwardly phrased because we did not combine the relational verb phrase and the relational noun phrase(s) into one component; however, for the remainder our correct extractions, the output was similar to, or better than the corresponding output of the BONIE model. A particularly interesting finding was that 42 of the 110 sentences we correctly extracted were not even attempted by the BONIE model.

The findings of the topic mining task were rather inconclusive. We found that both LDA and NMF had difficulty in clustering the data according to the classes we had assigned, even on the original text. NMF in particular often weighted terms in such a way that certain documents would have no single dominant topic. The only encouraging result came from NMF applied on a corpus consisting of only manually annotated relations, which suggests that perhaps such technique might work for document clustering. For topic modeling however, we only had inconclusive findings.

5.2 Future Work

The results of our rule-based numerical relation extractor are encouraging, however, there is clear room for improvement. As future work, we want to find new ways of detecting entity boundaries that do not rely on syntactic structure. Additionally, it is clear that our model needs a clearer rule set that is able to remove the large amounts of faulty extractions that it currently outputs, thus improving precision. Incorporating the verb into the relation component is also another improvement that our model requires, as this will enable it to output more natural sounding relations. The fact that only 62% of the correctly identified relations that our model output had a counterpart in BONIEs output is an interesting phenomena that definitely deserves further research. If it is possible to somehow incorporate some of our rules into BONIE, it might allow the system to output these extractions as well. Having a system that requires manual input of keywords, even if it is optional, is also against the idea of Open IE; as further work it would be interesting to search for alternatives to this strategy. Another major improvement to our model would be a way to standardize units, so that extractions from a corpus can be compared with other similar extractions without needing to worry about unit-of-measurement conflicts. A subset of numerical relations pose a problem that we did not get a chance to address in this thesis. Certain numerical relations take 2 entities alongside a quantity, for example “The distance between my home and the parking lot is 10m” should generate the tuple *(distance between, my home, and, the* *parking lot, is, 10m).* There are several other such relations that should be mapped out in a future work.

Due to the fact that free text, particularly informal text on forums or social media, can often violate proper grammar, it is clear that purely syntax based unlearning extractors will not perform well on what is a sizable proportion of all text. It even seems unlikely that a syntax based learning extractor will do much better; perhaps alternative or hybrid methods of finding extraction components is necessary for further development not only in numerical relation extraction, but also in the broader field of IE. Developing a standard evaluation technique for Open IE models that evaluates them more rigorously across all sorts of domains and all sorts of relations and removes the subjectivity of manual annotation of extractions is also an important task that will benefit all future Open IE studies.

Using extracted numerical relations for topic modeling still seems out of reach. As future work, we would like to greatly increase the number and length of documents in our test set. The documents chosen should feature a wide variety of topics, and should also consider how rich each topic is in numerical relations. Instead of ignoring the value of the quantity, if they are standardized down to a few select units, it might be possible to hash them in such a way as to match similar values of the same or similar relations across the corpus. Generally there is need for large amounts of labeled numerical relation data, and meeting this desire would go a long way to meeting several of our proposals.

Bibliography

[Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries - DL '00*, pages 85–94, San Antonio, Texas, United States. ACM Press.

[Allahyari et al., 2017] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Boutsidis and Gallopoulos, 2008] Boutsidis, C. and Gallopoulos, E. (2008). SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362.

[Brin, 1999] Brin, S. (1999). Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer.

[Chen et al., 2019] Chen, Y., Zhang, H., Liu, R., Ye, Z., and Lin, J. (2019). Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowledge-Based Systems*, 163:1–13.

[Etzioni et al., 2008] Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68.

[Fader et al., 2011] Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

[Habert et al., 1998] Habert, B., Adda, G., Adda-Decker, M., de Marëuil, P. B., Ferrari, S., Ferret, O., Illouz, G., and Paroubek, P. (1998). Towards tokenization evaluation.

[Hoffman et al., 2012] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2012). Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347.

[Hoffmann et al., 2011] Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. (2011). 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, Portland, Oregon, USA. Association for Computational Linguistics.

[Hudson, 2010] Hudson, R. A. (2010). *An introduction to word grammar.* Cambridge textbooks in linguistics. Cambridge University Press, Cambridge ; New York. OCLC: ocn636566718.

[Jónsson and Stolee, 2015] Jónsson, E. and Stolee, J. (2015). An evaluation of topic modelling techniques for twitter.

[Jung et al., 2012] Jung, H., Choi, S.-P., Lee, S., and Song, S.-K. (2012). *Theory and Applications for Advanced Text Mining: Survey on Kernel-Based Relation Extraction.* InTech.

[Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, N.J, 2nd ed edition. OCLC: 213375806.

[Kambhatla, 2004] Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* -, pages 22–es, Barcelona, Spain. Association for Computational Linguistics.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[Lee and Seung, 2001] Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

[Madaan et al., 2016] Madaan, A., Mittal, A., Mausam, Ramakrishnan, G., and Sarawagi, S. (2016). Numerical relation extraction with minimal supervision. In *AAAI*.

[Mahata et al., 2018] Mahata, D., Kuriakose, J., Shah, R. R., and Zimmermann, R. (2018). Key2vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 2 (Short Papers), pages 634–639, New Orleans, Louisiana. Association for Computational Linguistics.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press, New York. OCLC: ocn190786122.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing.* MIT Press, Cambridge, Mass.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). 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: Volume 2 - ACL-IJCNLP '09*, volume 2, page 1003, Suntec, Singapore. Association for Computational Linguistics.

[Newman et al., 2010] Newman, D., Lau, J., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. pages 100–108.

[Niklaus et al., 2018] Niklaus, C., Cetto, M., Freitas, A., and Handschuh, S. (2018). A survey on open information extraction.

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

[Qiang et al., 2019] Qiang, J., Zhenyu, Q., Yun, L., Yunhao, Y., and Xindong, W. (2019). Short text topic modeling techniques, applications, and performance: A survey.

[Quan et al., 2015] Quan, X., Kit, C., Ge, Y., and Pan, S. J. (2015). Short and sparse text topic modeling via self-aggregation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 2270–2276. AAAI Press.

[Saha et al., 2017] Saha, S., Pal, H., and Mausam (2017). Bootstrapping for Numerical Open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.

[Silge and Robinson, 2017] Silge, J. and Robinson, D. (2017). *Text mining with R: a tidy approach*. O'Reilly, Beijing ; Boston, first edition edition. OCLC: ocn993582128.

[Stanovsky and Dagan, 2016] Stanovsky, G. and Dagan, I. (2016). Creating a Large Benchmark for Open Information Extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

[Waegel, 2003] Waegel, D. (2003). A survey of bootstrapping techniques in natural language processing.

[Zelenko et al., 2002] Zelenko, D., Aone, C., and Richardella, A. (2002). Kernel methods for relation extraction. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, volume 10, pages 71–78, Not Known. Association for Computational Linguistics.