Numerical Relation Extraction with Minimal Supervision

Submitted in partial fulfillment of requirements for the degree of Master of Technology

by

Aman Madaan

Under the Guidance of Prof. Sunita Sarawagi, Prof. Ganesh Ramakrishnan and Prof. Mausam

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
June, 2015

To my family

Dissertation Approval Certificate

This dissertation entitled "Numerical Relation Extraction with Minimal Supervision", submitted by **Aman Madaan (Roll No: 133050004)** is Approved for the degree of **Master of Technology** in **Computer Science and Engineering** from **Indian Institute of Technology, Bombay.**

Examiners

____

____

____

Supervisors

____

____

____

Chairman

____

Date: ____________________ Place: IIT Bombay, Mumbai

Declaration

I, Aman Madaan, declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Aman Madaan, 133050004

Date: ____________________ Place: ____________________

Acknowledgement

Thanks to Prof. Mausam, Prof. Ganesh Ramakrishnan and Prof. Sunita Sarawagi for their support and guidance inspite of their busy schedules. Prof. Ganesh made sure that we were on the right track with the weekly meetings. His constant feedback and insights on the problem have been very helpful. Prof. Mausam has always lent a patient ear to our cribs on the model, extractions, feature weights and error analysis. His pin pointed advice on the road forward has been extremely crucial for our progress. Prof. Sunita, despite being on a sabbatical at Google Mountain View, always found time for guiding us. Inputs from her have played a major role in shaping the project to its current state.

A number of people form a support system that has helped my stay at IIT Bombay. I would like to thank Harshit Pande for the engaging discussions and the company. Thanks to Abhishek, Ashish, Harshit, Lucky, Ponraj, Praveen, Rashmi, Ruhi, Samson, Saransh, Someshwar for the outings.

I can't thank my family enough for supporting me in every decision I have ever made.

Abstract

The task of numerical relation extraction poses new, hitherto untackled challenges. The bewildering amount of false positives, units, modifiers are just some of the issues that are non existent for standard relation extraction, but become crucial when numbers are involved. We study these peculiarities in detail, and present two systems NumberRule and NumberTron, that perform extraction for numerical relations using some supervision.

NumberRule is our rule based systems that uses information in the dependency path between a number and an entity in conjunction with a small database of keywords for performing extraction.

NumberTron is a novel graphical model that uses a numerical fact database in and the keyword database used by NumberRule. Both the systems outperform a state of the art relation extractor, MulitR adapted for numerical relation extraction, with improvements of upto 33 F score points.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Massive knowledge bases containing the entire information of the web in a neat, ready to process, structured form continue to be a part of an IR researcher's reverie.

This is not unexpected; such knowledge bases have potential of revolutionizing the way in which information is searched by users on the web or exchanged by machines among themselves. Clearly, any progress towards a solution will have to deal with intricacies of how facts are expressed in the natural language. It turns out that such intricacies are too many to exhaust. Years of research has went into the aforementioned goal, and still there are plenty of loose screws. This report discusses one of them, numerical relation extraction.

We start with the problem definition, followed by an explanation of Distant supervision. We then present a summary of some of the key works. Subsequently, we discuss casting standard distant supervision for numerical relation extraction and the improvements achieved using Units and keywords. Finally, we discuss a simple rule-based relation extractor.

Terminology

- **a)Entity** An entity is something that exists in itself, actually or hypothetically. (**wikientity**)
- **b)Relation** A relation specifies a concept which binds two entities. For example, creator(Linux, Linus Torvalds).
- **c)Mention** A piece of text which expresses a relation. For example, the sentence “Linus Torvalds is the creator of Linux.”

- **d)Match** A mention expressing a relation R is called a match for R. The criteria for deciding whether the mention m is a match or not can vary. Presence of both the entities is, however, a mandatory condition.

- **e)Extraction** A 3-tuple $R(A, B)$ where A and B are entities related via a relation R.

Problem

Train extractors that can harness the Web for numerical relations, where relations are 3-tuples linking an entity to a number. For example,

- (India, **economy**, 1.842 trillion USD)
- (China, **internet users**, 590.56 million)
- (USA, **land area**, 2,959,054 square mile)

Chapter 2

Background

2.1 Learning From Unstructured Data

2.1.1 Snowballing

Let us motivate the idea by considering the following related problem:

Suppose we want to populate a repository of founders of companies, and all that we know is that Elon Musk is the founder of SpaceX. The problem can be divided into the following two parts, each of them rely on an intuition about how human beings form sentences in general.

- *Given an entity pair, and a corpus of documents, find out all the sentences that **express a relation** between the entity-pair.*

  Command line ninjas will quickly think of the following solution:
  grep -i 'entity1' sentences| grep -i 'entity2'

  The intuition behind this perhaps the most obvious solution is that *a sentence that houses both the entities can be expected to express a relation between them.* A quick web search with the query "entity1" and "entity2" will show that this intuition is not out of the blue.

- *Find what makes these sentences special*
  Sentence structure depends on the relation being expressed. In verbose, if two sentences express the same relation, there will be (okay, there can be expected to be) *features* that are similar in both of them. These include POS tags, words around the entities, dependency path between the entities to name a few.

Putting together the intuitions above, we can solve the problem as follows: Collect all the sentences which have SpaceX and Elon Musk in them, extract features from these sentences. Favor those features which repeat. Now use this set of features to extract similar pairs from other sentences. A fancier solution would be to reuse the extractions to learn more features, and continuing the process till the point of diminishing returns.

This seemingly shaky method actually works (Agichtein and Gravano 2000) and is popular by the name of snowballing.

2.1.2 Distant Supervision

Introduction

The basic setup is as follows:

- **a) KB:** A knowledge base consisting of facts. The facts are 3-tuples; the entities and the corresponding relation. For example:

Entity	Entity	Relation
Donald Knuth	Wisconsin	Born In
Srinivasa Ramanujan	Erode	Born In
Alan Turing	London	Born In
Alon Musk	SpaceX	Founder Of

  has 4 different facts

- **b) Corpus** The repository of text where we expect to find the sentences that express facts that we know. We need another repository, called the test set, where we will run our extractor to obtain new facts. These two can be the same.

Distant Supervision Assumption

Every sentence that has an entity pair (e_1, e_2) expresses the relation which exists between (e_1, e_2).

Matching

We next need to align our knowledge base with the corpus. This process is also called matching.

Algorithm 1 Distant Supervision: Matching

```
Corpus C, Knowledge Base KB
Training data, D, A set of matches Break C into a set of sentences, S
for each sentence s in S do
    E = all entity pairs in s
    for each entity pair ($e_1$, $e_2$) in E do
        if ∃ relation r in KB with r($e_1$, $e_2$) then
            add s to D with label r
        end if
    end for
end for
```

Training

Recall that obtaining the sentences which express a relation gives us training data, which we want to use to learn relation extractors, our goal. There are several ways to achieve this, starting from the naive ways of training sentence level classifier extending to fancier graphical model based learning. There are broadly two different ways of used in the literature for training extractors using the distant supervision data. Earlier work on distant supervision used the training data to train classical classifiers, like naive bayes, LR and so on. Recent work has shifted towards modeling extraction using graphical models.

Chapter 3

Related Work

3.1 Numbers in NLP

We discuss two recent works that are explore the role of numbers in the natural text.

3.1.1 Reasoning About Quantities in Natural Language

(Roy, Vieira, and Roth 2015) develop techniques supporting quantity semantics by introducing the problem of quantity entailment: Given a hypothesis h (a quantity, in this case) and a piece of text T, the problem involves determining whether or not T entails (or explains) the quantity h . The work defines a quantity expressed in the text as a triple *(v, u, c)* where v is value, u is the corresponding unit and c captures the kind of change that is happening in the quantity (increase, decrease) The parameter c however only serves as a flag that indicates whether or not the quantity is relative. The definition of quantity does not provide for storing the value of this change.

Quantity Identification They introduce a two step quantity identification system: the first step is a statistical boundary identifier, followed by a set of rules that derive the quantity as defined above from the segmented phrase. The set of rules include conversion to the standardized units (e.g. miles to meters), rewriting to known units (USD to US$), replacing non scientific units with the synsets (Israelis to people) and so on. The paper also hand waves the method of generating *implicit quantities* like *12 people* from *6 Korean couples.*

Quantity Comparison The quantity comparison algorithm takes two quantity-value triples, one from text T and the hypothesis, and returns whether T entails h, contradicts it, or has no relation whatsoever with it. The algorithm looks for compatible units and change phrases. The values are compared using the subset definition.

Quantity identification and comparison are then used together to answer the question on entailment as defined above.

The work also describes an attempt at automatically solving simple word problems (problems involving 2 or 3 quantities, and operators like addition, subtraction, multiplication and division). They solve this problem by using a cascade of classifiers.

3.1.2 Extraction and Approximation of Numerical Attributes from the Web

(Davidov and Rappoport 2010) explore the problem of extracting attributes like height and weight from the text. They use a set of 10 hand crafted patterns to retrieve three kinds of information about attributes of objects. The first is the absolute value, like *Obj width is *[width unit].* The next is the boundary information, like *the widest [label] is [width units]* bounds and finally comparison information, like *[Object1] has the same width as [Object2].* They next use a snowball (Agichtein and Gravano 2000) like approach to iteratively extract more patterns and object-attribute information. This information is finally arranged in a graph, which is used to answer the queries. Values for missing attributes are obtained via interpolation using the ordering information as obtained from the comparison patterns.

3.2 Rule Based Relation Extraction

In a paper titled *Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!*, (Chiticariu, Li, and Reiss 2013) make a strong case for the rule based system. We present two excerpts from the work that capture the gist of the very empathetic argument made in the paper.

> *If current trends continue, the business world will move ahead with unprincipled, ad-hoc solutions to customers' business problems, while researchers pursue ever more complex and impractical statistical approaches that become increasingly irrelevant*

> *"Even in their current form, with ad-hoc solutions built on techniques from the early 1980's, rule-based systems serve the industry needs better than the latest ML techniques."*

They present figure 3.1 to support their claims.

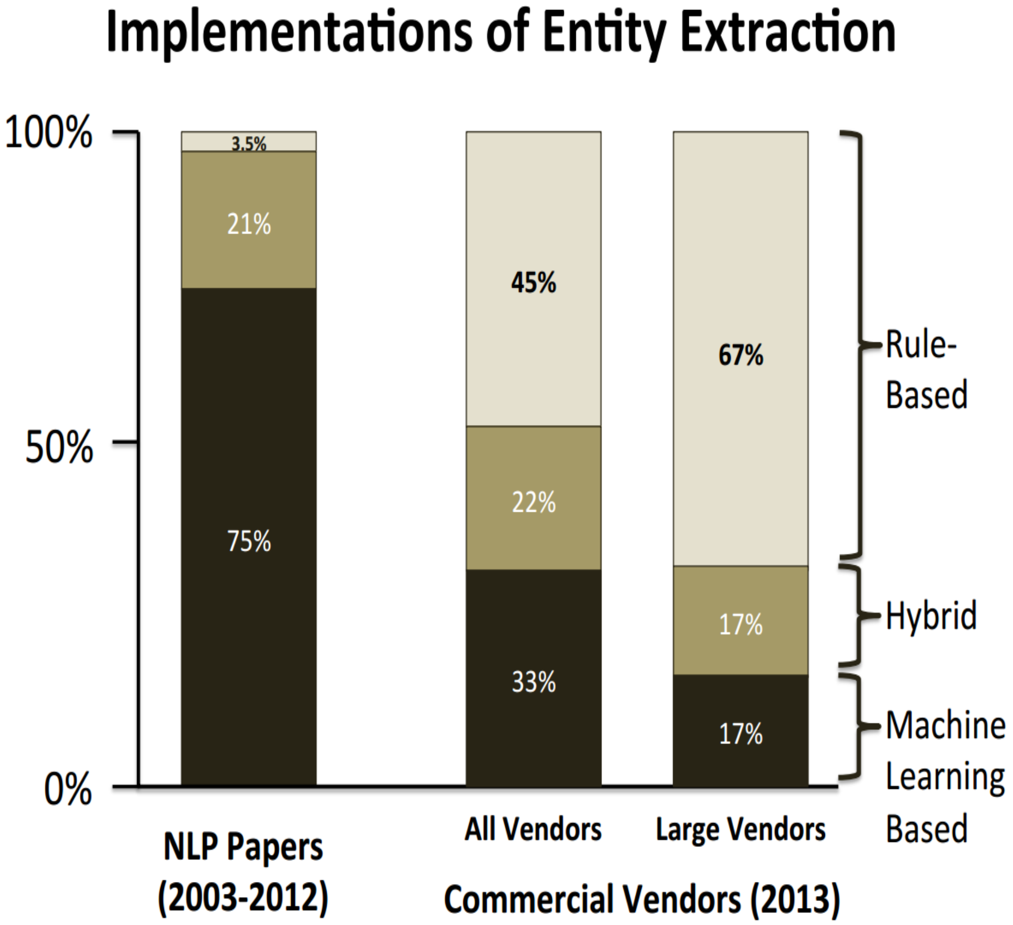


Figure 3.1: Popularity of Rule Based Systems: Academia vs. Industry

As the figure shows, for the problem of entity extraction, the machine learning based systems dominate the academia, whereas the rule based systems that are the ones grinding the gears on ground zero. We next discuss some rule based information extraction systems. The works are selected to maximize the coverage in terms of the popular concepts involved in the rule based extractions.

3.2.1 Automatic Acquisition of Hyponyms from Large Text Corpora

(Hearst 1992)

Start by presenting a way of learning new hyponymns relations from free text by using a list of hand compilked systactic patterns. The list of patterns include those like *such NP as NP ,* (or — and) NP* . The authors propose six patterns in total.

When applied to a sentence like *works by such authors as Herrick, Goldsmith, and Shakespeare.*, this will yield the following extractions. hyponym("author", "Herrick"), hyponym("author", "goldsmith"), hyponym("author", "Shakespeare")

The authors also present a method for learning new patterns. The algorithm they outlined involves fixing on the target relation, finding some entity pairs for which the relation is known to exist, find sentences where these entities appear, finding the commonalities in such sentences *(learning features!)* to construct patterns and then repeating the process with the patterns just obtained. It can be argued that they in fact suggested what (Craven and Kumlien 1999) refined as distant supervision as discussed in the background section and elaborated later in this chapter. The authors were able to find all the 6 hand crafted patterns using this technique. (Abacha and Zweigenbaum 2011) apply similar ideas for extracting relations between medical entities.

The authors next discuss adding the extractions to an existing knowledge base such as the wordnet. The challenges involve resolving ambiguity in the sense of either of the entities, and the case of missing entities.

3.2.2 Relation Extraction Using Parse Trees

(Fundel, Küffner, and Zimmer 2007) present RelEx, a system aimed at extracting relations in the medical domain that relies on the information provided by the dependency parse tree. The system first extracts a dependency path that might contain information useful for relation extraction. This is done using four rules. The rules impose constraints on the words present and the types present in the dependency path. For example, one of the rules they use reads: ***"Rule 3 extracts two noun phrase chunks connected by a dependency of the type between provided that the successor in the tree contains the word and or has a dependent noun phrase chunk, which is connected via an and dependency***. They expect the entities to be recognized using a list. Extractions undergo an additional *negation check* (ignoring an extraction if a candidate or a successor contains a negation word like *no, not, nor, neither*) and those that fail it are ignored.

The relation extraction starts after the dependency paths have been extracted. They use a list of keywords pertaining to different relations, and the presence of a keyword in a dependency path is an indicator of the particular relation being present. The authors call this list of keywords *relation restriction terms*. The system retains all the keywords for which even one restriction term (keyword) is present.

3.2.3 Rulebased Extraction of Spatial Relations in Natural Language Text

(Chunju Zhang et al. 2009) compile over 50 hand crafted patterns to detect spatial relations (distance, directional and topological) between geographical entities. For example, *entity1* is located in *entity2*, *entity1* is located north of *entity2* and so on.

Some of the example patterns are as follows *"geoNE1 + Direction Word + Verb + geoNE2"*,*"geoNE1 + Direction Word + V + geoNE2 + quantifier + unit"*, *"geoNE1 + V + geoNE2 + Direction Word + quantifier + unit"* and *"geoNE1 + Verb + geoNE2 + Direction Word"*.

3.3 Distant Supervision: History and some recent works

The first distant supervision paper came out in 1999 (Craven and Kumlien 1999). The process has remained more or less the same since the first paper, with different works stemming out of either relaxations of different assumptions made by (Craven and Kumlien 1999) or by using a smarter matching algorithm.

3.3.1 Constructing Biological Knowledge Bases by Extracting Information from Text Sources

This was the first work to use distant supervision for creating a repository of biological facts. They targeted 5 different relations between Proteins, Tissues, Cell-Types, Diseases, Pharmacologic-Agents and Subcellular-structure. A naive bayes based simple relation extractor is first described. The extractor works in 2 steps: A classifier trained on hand labeled data first labels whether a sentence *can* express a particular relation. If yes, the sentence is searched for a pair of one of the 5 types depending on the label and the fact is added to the database. The authors note that it took an expert 35 hours to hand label the corpus. This forms the basis of motivating distant supervision based methods.The paper also identifies many areas of improvements, like a pair of entities taking up multiple relations, on which subsequent work has been built up.

The authors obtain an improvement of around 9 precision points with distant supervision. It is possible that the following points contributed to the improvement in scores:

- Constrained entity set. Proteins and Tissues won't just appear together without any possible relation.

- The corpus used was very well aligned with the knowledge base.

3.3.2 Distant Supervision for the Web

(Mintz et al. 2009) revived the interest of the community around the problem of distant supervision. The major contributions of this paper can be listed as follows:

- **Distant Supervision for the Web** Craven and Kumlien had a limited knowledge base and a limited corpus to align it with. As the web exploded, the knowledge bases that became available were of the magnitude of FreeBase, and the corpus that can be aligned with it was the web. Mintz et. al bought this fact to the spotlight and sparked a series of works.

- **Features** They designed a set of rich features that are used by researchers to date. These features include dependency paths, POS tag sequences, word sequences to name a few.

3.3.3 Learning 5000 relation extractors

(Hoffmann, Congle Zhang, and Weld 2010) The number of relations typically used in the works were limited to the range of 30-40. This work targeted 5000 relations, which is a big jump. To fight sparsity that will arise in the training data due to a large number of classes, they also train a top level classifier which decides which extractor should be used for a particular document. Not all the extractors are employed for a given document.

3.3.4 Modeling Relations and Their Mentions without Labelel Text

Distant supervision assumption does not really holds when the knowledge base is not well aligned with the corpus. Another way of saying this would be that the corpus can be expected to consist of sentences that can host entity pairs in a wide range of contexts, and the sentences may have nothing to do with the relations in which the entity pairs appear. They take the example of the relation "nationality". A wide range of popular entities that are born in the country will be mentioned in the sentences that may have nothing to do with the fact that they were born in that country. In such cases, it can be argued that the training data generated will be extremely noisy and will lead to poor extraction performance. (Riedel, Yao, and McCallum 2010) relax the distant supervision assumption, not every sentence mentioning the entity pair will express a relation that is present in the knowledge base between the two entities. The constraint of one relation per entity pair still holds though.

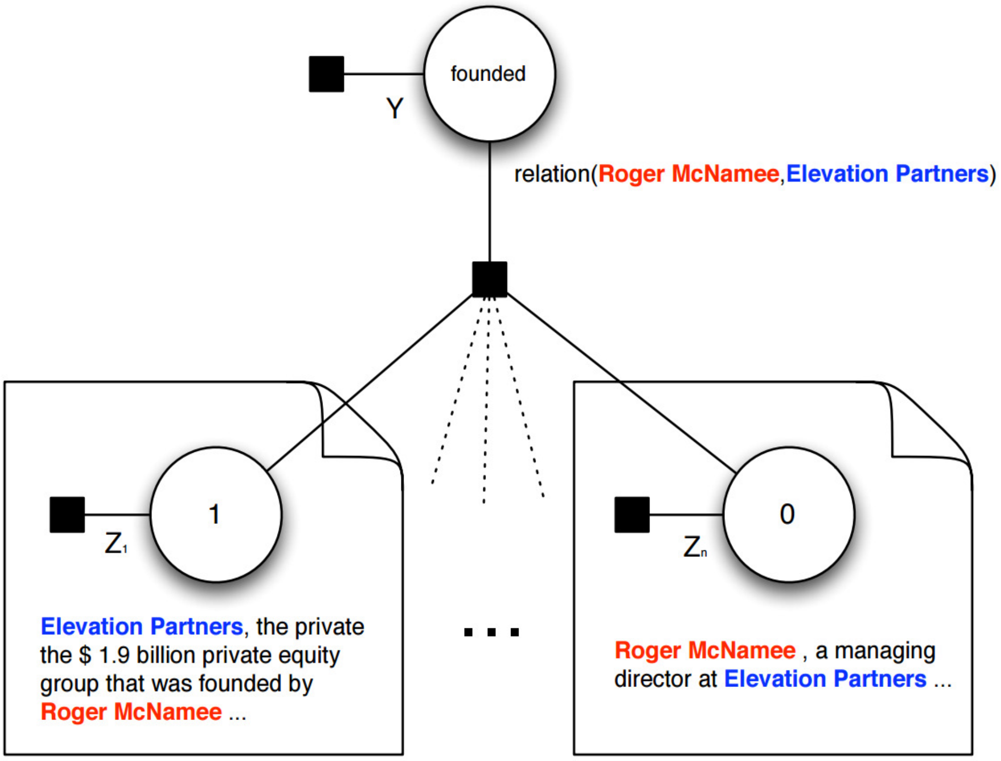
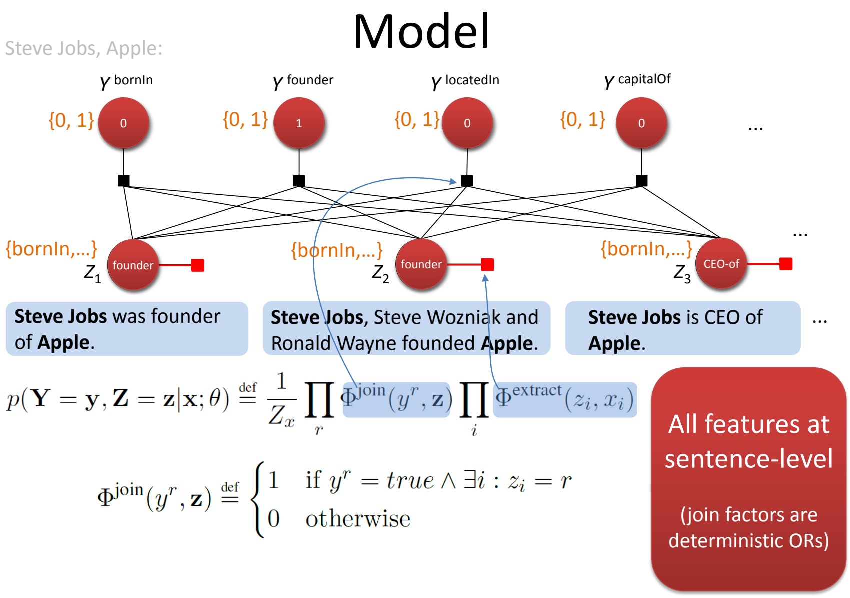

founded
Y
relation(Roger McNamee,Elevation Partners)
1
Z₁
Elevation Partners, the private the $ 1.9 billion private equity group that was founded by Roger McNamee ...
...
0
Zₙ
Roger McNamee , a managing director at Elevation Partners ...

Figure 3.2: Riedel et. al: Relaxing the Distant Supervision Assumption

Riedel et. al were also the first to introduce graphical models into the distant supervision world to the best of our knowledge. Figure 3.2 shows the graphical model used by them. The role of the Z variables remains the same as it is in the models discussed below. We note that the model is unable to capture multilable scenario: every entity pair is still tied to just one label.

3.3.5 Knowledge-based Weak Supervision for Information Extraction of Overlapping Relations (MultiR)

Some of the entity pairs can occur in multiple relationships. For example, an athlete usually *plays for* the country they are *born in*. The *CEO of* a company also *works for* it. The techniques discussed till now are focused on extracting only one of the relations for an entity pair. MultiR by (Hoffmann, Congle Zhang, Ling, et al. 2011) relaxes one relation per entity-pair assumption. (Hence the name, multi-relation). MultiR achieves this by modeling both sentence level and corpus level extraction decision. The graphical model is as shown in figure 3.3 (taken from (Hoffmann, Congle Zhang, Ling, et al. 2011))



Model

Steve Jobs, Apple:

Steve Jobs was founder of **Apple**.

Steve Jobs, Steve Wozniak and Ronald Wayne founded **Apple**.

Steve Jobs is CEO of **Apple**.

$$p(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z}|\mathbf{x};\theta) \overset{\text{def}}{=} \frac{1}{Z_x}\prod_r \Phi^{\text{join}}(y^r,\mathbf{z}) \prod_i \Phi^{\text{extract}}(z_i,x_i)$$

$$\Phi^{\text{join}}(y^r,\mathbf{z}) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } y^r = \mathit{true} \wedge \exists i : z_i = r \\ 0 & \text{otherwise} \end{cases}$$

All features at sentence-level

(join factors are deterministic ORs)

Figure 3.3: MultiR Graphical Model

There are two different types of nodes in the model:

- **a) Sentence Level extractions, Z:** The cardinality of these variables is equal to the number of targeted relations + 1 for "no attachment", which means that the sentence expresses no relation. This is important for relaxation of the DS assumption.

- **b) Corpus level extraction, Y:** The Ys are the binary variables, one for each relation. Y_i takes the value of 0 or 1 depending on whether the *ith* relation is expressed in the corpus or not.

There are 2 factors:

- **a) Factors local to Z** These factors capture the likelihood of a sentence belonging to one of the relations. The usual technique of representing factor tables in log-linear forms is used. The features are as defined by (Mintz et al. 2009).

- **b) Factors between Y and Z** These factors ensure that the probability mass is divided only amongst the possible extractions. As shown in the figure, the factor takes a value of 1 only if the corpus level bit is 1 i.e. that relation exists at corpus level **and** at least one of the sentences also expresses it.

3.3.6 Modeling Missing Data in Distant Supervision for Information Extraction

(Ritter et al. 2013) extend MultiR to handle the case of missing data. The data could be missing from the database (the knowledge base) or the text. They achieve this by adding another layer of random variables above the aggregate level y nodes in the multiR, and call it the d nodes as shown in figure 3.4 [1].

The learning objective is designed to benefit the configurations in which the d and the t nodes are in agreement. The full inference is again straight forward as in the case of MultiR. The conditional inference problem no longer reduces to the edge cover problem, and is solved by doing a local search using A*.

[1] Taken from Ritter et al. 2013

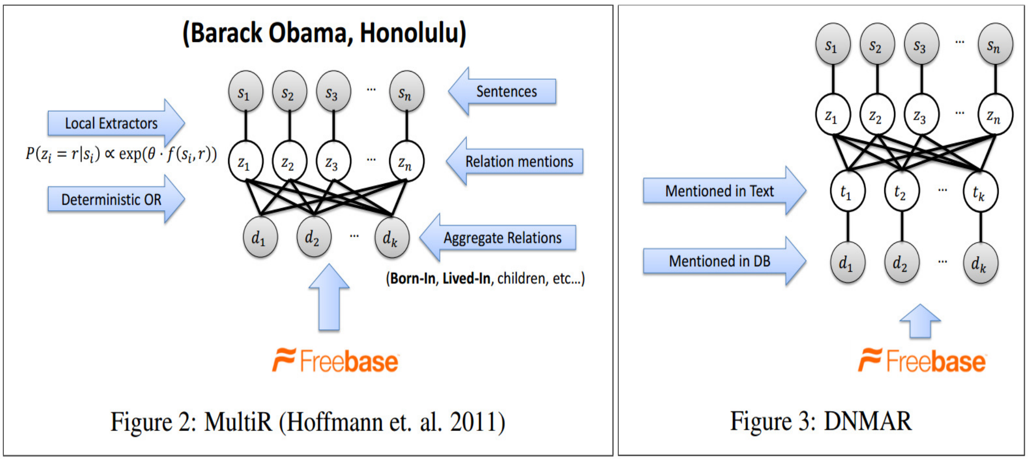


Figure 2: MultiR (Hoffmann et. al. 2011)

Figure 3: DNMAR

Figure 3.4: DNMAR: MultiR Extended to Handle Missing Data

3.3.7 Type-Aware Distantly Supervised Relation Extraction with Linked Arguments

(Koch et al. 2014) investigate several tweaks that are possible in the distant supervision pipeline. They add two improvements in the matching step of the distant supervision pipeline: use of a named entity disambiguator over a simple named entity tagger, and the use of coreference resolution in detection of arguments.

They also study the use of entity types during training and extraction. They train based on types and then add restriction on types during extraction (*PERSON-PERSON* cannot be involved in the relation *Born in*) This is exactly similar to adding unit based constraints for numerical relation extraction.

3.3.8 Multi-instance Multi-label Learning for Relation Extraction (MIML)

(Surdeanu et al. 2012) relax the same set of assumptions as (Hoffmann, Congle Zhang, Ling, et al. 2011), but differ in two ways. First, they do not flip the aggregate latent variables (ys in the case of Multir) deterministically, and instead use a classifier that can learn constraints that can prevent some of the aggregate variables from being turned on together. The second difference is the training method; while MultiR relies on the perceptron like training algorithm (Collins 2002), MIML uses EM algorithm to learn weights for the mention level and instance level classifier.

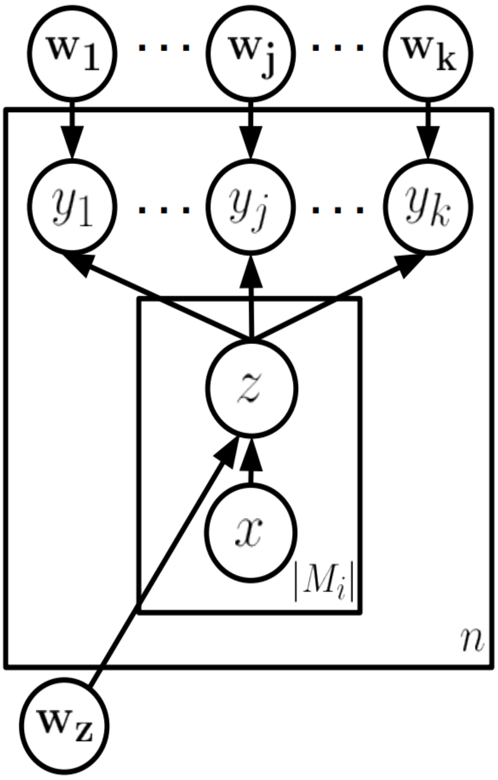


Figure 3.5: The MIML Graphical Model

Because of its similarity with MultiR, we present the graphical model next followed by an intuitive explanation of the training algorithm[2].

Figure 3.5 shows the miml graphical model. The Z nodes capture the sentence level (latent) relation label. The values that the z nodes can take range from one of the relations and the NA class. The y variables indicate the aggregate level relations that are present among the entity pair i. $w_1, w_2, ..., w_k$ are the aggregate level binary classifiers that are used to make a decision on an individual aggregate node. The features used are the

w_z is a multiclass classifier that assigns one of the relation labels or the label NA to the mention level variables, Z.

3.3.9 Infusion of Labeled Data in Distant Supervision

(Pershina et al. 2014) considers the setting in which we have a small amount of high quality data and a large, unlabeled corpus. The idea is to learn good features from the small amount of high quality data, and to give these

[2] figure taken from (Surdeanu et al. 2012)

features higher priority while training the extractor in the standard distant supervision setting.

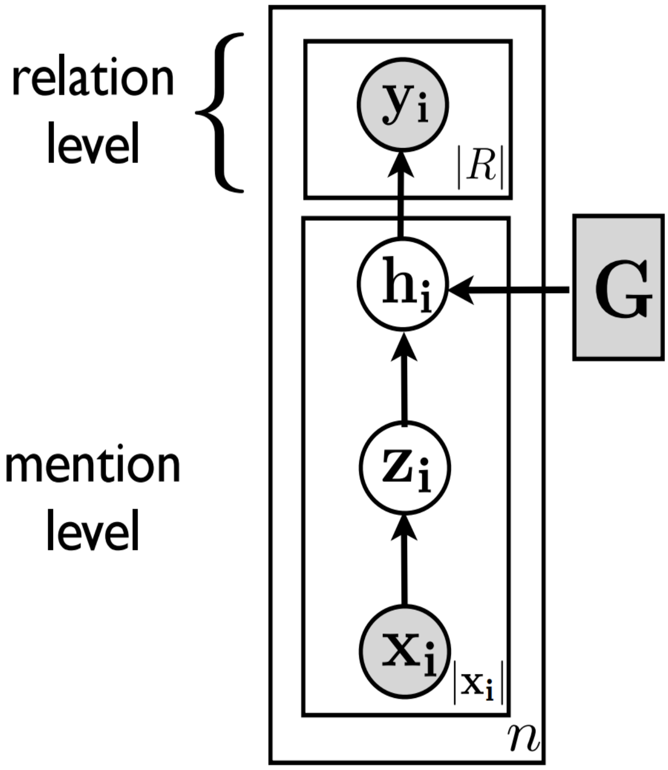


Figure 3.6: The MIML Graphical Model Modified to Infuse Hand Labeled Data

As shown in figure 3.6 [3], the h nodes override the decision made by the instance level classifiers. G refers to the set of feature-relation mapping learned from the hand labeled dataset. If any of features fired in the sentence x_i belongs to one of the rules in G, h is flipped to take the corresponding label. Otherwise, the label assigned to Z_i via the instance level multi-class classifier is copied to h_i.

3.3.10 A Convex Relaxation for Weakly Supervised Relation Extraction

Most of the works presented in the previous subsections are based on graphical models. We now present an orthogonal approach for relation extraction

[3] taken from (Pershina et al. 2014)

using distant supervision based on convex optimization. Following Bach and Harchaoui, (Grave 2014) uses linear classifier W $\in \mathbb{R}^{D\times(K+1)}$, the squared loss and the squared l_2-norm as the regularizer. Thus, their initial formulation of the primal problem becomes:

$$\begin{aligned} \min_{\mathrm{Y,W}} \quad & \frac{1}{2}||\mathrm{Y}-\mathrm{XW}||_F^2+\frac{\lambda}{2}||\mathrm{W}||_F^2, \\ \text{s.t.} \quad & \mathrm{Y} \in \{0,1\}^{N\times(K+1)}, \\ & \mathrm{Y1} = 1, \\ & (\mathrm{EY}) \circ \mathrm{S} \geq \mathrm{R}. \end{aligned} \tag{3.1}$$

where X $\in \mathbb{R}^{N\times D}$ be the feature matrix representing the relation mention candidates. The closed form solution of the matrix W is given by

$$\mathrm{W} = (\mathrm{X}^T X + \lambda \mathrm{I}_D)^{-1}\mathrm{X}^T\mathrm{Y} \tag{3.2}$$

Putting this value of W into the objective function of the optimization problem (3.1), applying woodbury matrix identity and adding slack variables to the objective, the final primal problem becomes:

$$\begin{aligned} \min_{\mathrm{Y},\xi} \quad & \frac{1}{2}\mathrm{tr}(\mathrm{Y}^T(\mathrm{XX}^T+\lambda\mathrm{I}_N)^{-1}\mathrm{Y}) + \mu||\xi||_1, \\ \text{s.t.} \quad & \mathrm{Y} \geq 0, \quad \xi \geq 0, \\ & \mathrm{Y1} = 1, \\ & (\mathrm{EY}) \circ \mathrm{S} \geq \mathrm{R} - \xi. \end{aligned}$$

It might be interesting to note how the different constraints that were present in the graphical model based systems presented so far have been formulated here.

$\mathrm{Y1} = 1$ state that every mention should have at most one label.

$(\mathrm{EY}) \circ \mathrm{S} \geq \mathrm{R} - \xi$ ensure that for every fact that is present in the database, there should be at least one mention. The use of the slack variable indicates that we don't really trust our database or the corpus.

Chapter 4

Peculiarities of Numerical Relations

Many of the challenges in numerical relation extraction stem from the fact that numbers have no identity of their own; they represent count of some real entity or phenomenon. The amount of false positives that it leads to is unprecedented.

4.1 Numbers and False positives

Why would numbers lead to false positives in the first place? The problem stems from the fact that numbers have no identity of their own; they represent count of some real entity or phenomenon.

- **Numbers can appear in many more contexts with an entity** The number of ways in which any two entities can appear together in a sentence is far less than the number of ways in which a number and a quantity can appear together. For example, Consider the entity pair "Bill Gates" and "Microsoft" and the entity-number pair "Bill Gates" and "3" (say). While former will usually co-occur in finite contexts (Founder, CEO, Evangelist etc.), the latter may co-occur anywhere Bill Gates happen to be around something which is 3, the number of cars, billion dollars donated, number of units headed, position in the company, number of business units shutdown by Microsoft and so on.
- The situation is worse for smaller whole numbers, which are more frequent. This intuitively makes sense as we are more often see 2,3 or 11 than 111212233 or 11.42143.

- **The match mines**

  During the initial phases of our experiments, we stumbled upon the *match mines* These were basically huge tables, world cup scores of all the matches played and so on. A couple of such sentences were responsible for 21% of the matches! It is easy (and very important) to get rid of such sentences. For subsequent runs, we first sort the sentences by length and then remove top 1000 of them.

4.2 Numbers are weak entities: a case for keywords

Matches from unit extraction showed that in some cases, the sentence that supposedly labeled as a match for a particular relation has no mention of the relation itself at all. For example, consider: *In eurozone powerhouse Germany, industrial orders jumped 3.2 percent in June, official data showed Thursday, with foreign demand behind a sharp rebound following a surprise drop in May.* In this sentence, (Germany, 3.2) was considered as a match pair for the relation Internet user percent. Clearly, it has nothing to do with it.

4.3 Numerical Relations are Explicit

A key observation that can be made by going through the sentences that express numerical relations is that one cannot be too poetic while forming a sentence that is supposed to state a numerical fact. This is in stark contrast with sentences expressing relations between entity pairs, wherein the underlying relation might be implicit. If we want to state GDP of a country in a sentence, there is no escape from the words like “GPD” or “gross domestic product” and the likes * .

Compare this with a sentence that must relate Microsoft and Bill Gates. A few ways of stating that Microsoft was founded by Bill Gates can be enumerated as follows:

- Bill Gates is the founder of Microsoft
- Bill Gates founded Microsoft
- Bill Gates is the father of Microsoft
- Bill Gates laid the foundation stone of Microsoft

- Bill Gates started Microsoft

If this is indeed true, imposing an additional constraint of keyword being present in a sentence in addition to the fact being present can help in cutting down the number of false positives. We note that such a pruning is possible only in case of numerical relations. As mentioned earlier, for real world entity pairs, co-incidental matches will be rarer and a constraint on the relation word being present will be too restrictive.

4.4 Numbers change

In case of non numerical relations, both the arguments usually never change. Numbers, however, change all the time. Population, inflation, internet user percentage, height are some of the examples that either change or occurr with variations with the same entity (height of a giraffe). This further prevents hard comparisons of the numbers present with the entries in the knowledge base.

4.5 The facts may not be reported in absolute terms, but the change might be reported

For numerical quantities, a change is as interesting as the real value. Notable examples include GDP and inflation. We do not extract such changes, but handle such cases by using a set of hand crafted delta words. For example, *Manteca expects its population to increase by about 44,000 by the year 2035.*

4.6 Units: The types for Numbers

Real world entities can be assigned types at various levels of granularities: coarse grained NER types to fineNamed entity disambiguation, see Sarawagi 2008). Units act as a type for numbers, and thus a unit tagger becomes crucial for a system dealing with numbers.

Chapter 5

NumberRule

We now present a rule based system, called NumberRule that uses information on units and a handful of keywords to extract numerical relations from text.

5.1 The Shortest Path Hypothesis

From (Bunescu and Mooney 2005),

If e_1 and e_2 are two entities mentioned in the same sentence such that they are observed to be in a relationship R, our hypothesis stipulates that the contribution of the sentence dependency graph to establishing the relationship $R(e_1, e_2)$ is almost exclusively concentrated in the shortest path between e_1 and e_2 in the undirected version of the dependency graph.

This *shortest path hypothesis* states that the evidence that would be indicative a relation can be expected to be present in the dependency path for majority of the cases.

As discussed in chapter 4, *keywords* can act as the only piece of evidence that is really needed in the case of numerical relations. We also need to be aware of *modifying relations*; those sentences that do not report an absolute fact but only report a change.

5.2 Dependencies

Dependencies are grammatical relation between two words, governor and dependent. The relation captures the way in which one of the words is affected by the other.

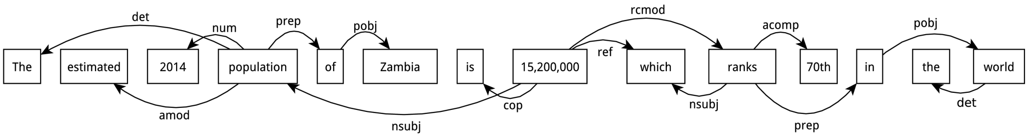


Figure 5.1: Dependency Graph

For example, consider the sentence: "The red ball was lost" The dependencies are:

- **amod(ball,3,red,2)** "Red" is an adjective for "ball"
- **det(ball,3,The,1)** "the" is a determiner of "ball"
- **nsubjpass(lost,5,ball,3)** "ball is the subject of lost"
- **auxpass(lost,5,was,4)** "was is an auxiliary of lost"

5.3 Dependency Path

We define a dependency path between two words "A" and "B" as the shortest path between them in the dependency graph. The dependency graph consists of one node for each of the words, and the dependencies are the collapsed typed dependencies as obtained from the stanford dependency parser (Manning et al. 2014). Figure 5.1 shows dependency graph for *"The estimated 2014 population of Zambia is 15,200,000, which ranks 70th in the world."*

5.4 Keywords, Delta Words, Modifiers

- **Keywords** Sentences expressing numerical relations can be expected to be explicit about the relation being expressed. Stated another way, we can expect presence of certain keywords that might help in identifying relations.
- **Delta words** A large number of false positives stem out of mentions where a change in the numerical attribute is mentioned. We detect such cases using a set of delta words, some of which are as shown in table A.2
- **Modifiers** A word m is said to be a modifier of the word w if there is a modifying dependency from m to w, like a ***blue*** *whale* or ***urban*** *population.*

5.5 Augmented phrases

Let S be a sentence and W be a word in the sentence. The *Augmented Phrase* W' is formed by concatenating W with words P such that W and P are related via a *modifying dependency.*

5.6 The modified shortest path hypothesis

With definition of augmented phrases as given, we present the modified shortest path hypothesis as follows:

If L is a location and N is a number mentioned in the same sentence S, ***and*** *if the shortest path between L and N in the undirected version of the dependency graph of S consists of a keyword indicative of a numerical relation R,* ***and*** *has no word indicating a change in the numerical quantity,* ***and*** *unit of N is compatible with R, then the sentence expresses the relation* $R'(L', N)$ *where* L' *is the augmented location phrase and* R' *is the augmented relation phrase.*

5.7 Numerule Relation Extraction

NumberRule relation extraction algorithm based on the modified shortest path hypothesis is as presented in Algorithm 2.

5.7.1 Example

Consider the sentence: *"The estimated population for 2014 of the Australian continent is about 36.25 million people"*

The shortest path between the Country-Number pair (Australian, 36.25) will be:

Australian$\xrightarrow{amod}$continent$\xrightarrow{prep_of}$2014 $\xrightarrow{prep\_for}$*population* $\xrightarrow{nsubj}$ people$\xrightarrow{num}$million $\xrightarrow{number}$36.25

The path between Australia and 36.25 has the keyword *population*, so this will lead to an extraction POP(Australian, 36.25 million).

If we modify the sentence to: *"The estimated population for 2014 of the Australian continent increased by about 3.25 million people"*

The shortest dependency path now becomes:

Australian$\xrightarrow{amod}$continent$\xrightarrow{prep_of}$2014 $\xrightarrow{prep\_for}$*population*$\xrightarrow{nsubj}$ increased$\xrightarrow{prep_by}$people$\xrightarrow{num}$milli $\xrightarrow{number}$36.25

Algorithm 2 NumberRule Relation Extraction

```
Given a sentence S, let $E_S$ be the set of entities and $N_S$ be the set of numbers that are present in the sentence.
Let $R$ be the set of relations
Let $LegalUnits(r)$ be the set of possible units for a relation $r$ in $R$
Let $Unit(n)$ be the unit for a given number $n$ as returned by the unit tagger.
Let $K$ be the set of keywords (Table A.1)
Let $\Delta$ be the set of delta words (Table A.2)
for $(e, n) \in (E_S \times N_S)$ do //For all entity-number pairs
    $P \leftarrow$ the set of words in the dependency path between $e$ and $n$
    for $r \in R$ do
        if $P \cap K_r = \emptyset$ then //keyword is not present
            continue;
        end if
        if $P \cap \Delta \neq \emptyset$ then //delta words are present
            continue;
        end if
        if $Unit(n) \notin LegalUnits(r)$ then //incompatible units?
            continue;
        end if
        Extract $r(e', r', n)$, where e' and r' are the augmented entity and relation phrases.
    end for
end for
```

The path will additionally have a modifier, “increased”, and thus there won’t be any extraction.

As an example of the importance of using augmented phrases, consider the sentence:

As another example, consider “Female population of urban india is 23 million”. The shortest path between the arguments (India, 23) is (Population, million) and thus a vanilla system will yield the wrong extraction **POP(India, 23 million)**, instead of the correct **POP(urban india, female population, 23 million)**.

5.8 Improving Precision and Recall

We now list some secret sauce that further boosted the precision and recall of our rule based system. Note that all these fixes are motivated by real world examples, and are not quick patches to overfit the test data.

5.8.1 Ignoring modifiers preceded by *to*

The earlier version ignores extractions if the path has any modifier like increase, decrease, rose etc.. The idea was to ignore sentences that only communicate change in quantities. For example, *The internet user percent has increased by 3.4 over the last year due to telecom reforms.*

However, vanilla implementation of this idea leads to false negatives. Consider: *The population of India **increased to** 1.3 billion.*

Thus, presence of *to diffuses* the modifiers in current version.

5.8.2 Keywords modifying words on the shortest path

There were some cases in which the keywords were not on the shortest path, but instead modified words that were present on the shortest path, we consider such extractions.

5.8.3 Handling Redundant Extractions and Spurious pairs

For a given augmented entity phrase E, and a given augmented relation phrase R, Let

- N_1, N_2, ..., N_i be the numbers such that modified shortest path hypothesis leads to extractions $R_C(E, R, N_1)$, $R_C(E, R, N_2)$, ... , $R_C(E, R, N_n)$ where R_C is the relation code.

- Let $|E, N_i|$ be the shortest distance from the central entity of the entity phrase to the *ith* number.
- Let $|E, N_k| \leq |E, N_i| \; \forall i \in [1, n]$.

Then, we ignore all the extractions except $R_C(E, R, N_k)$.

An example is the sentence *India's population sits at 1.3 billion, whereas Pakistan's population has been hovering around 1.1 billion.* Which now leads to correct extractions:

- POP(pakistan, population, 1.1 billion)
- POP(india, population, 1.3 billion)

5.8.4 Ignoring Dates

Clearly, only matching with numbers will attract entities like dates that we are not interested in. We ignore all the numbers that are marked as being a part of date.

Chapter 6

NumberTron

In this chapter, we discuss an alternative implementation of the NumberTron, the one in which the Z nodes are multiary and not binary. We start with a description of the graphical model, followed by a discussion on the features and the training algorithm. We finally discuss the sentence level extraction level that we follow.

6.1 The Graphical Model

NumberTron creates a graph for each entity. We list the necessary definitions before proceeding with the description of the graphical model.

6.2 Definitions

For an entity e

- Let S_e be the set of sentences that express the entity e.
- Let Q_e denote the distinct numbers with unit that appear in footnote. We use the unit tagger by Sarawagi and Chakrabarti 2014 to identity units of numbers in the text and to convert all unit variants like "mile", "km" to a canonical SI unit, "meter".
- $\forall q \in Q_e$, let $S_{e,q} \subseteq S_e$ denote the sentences that mention e and q.

6.3 Random Variables

For each entity e and relation r, our graphical model contains a binary random variable n_q^r for each $q \in Q_e$ denoting if the number q is related to e via

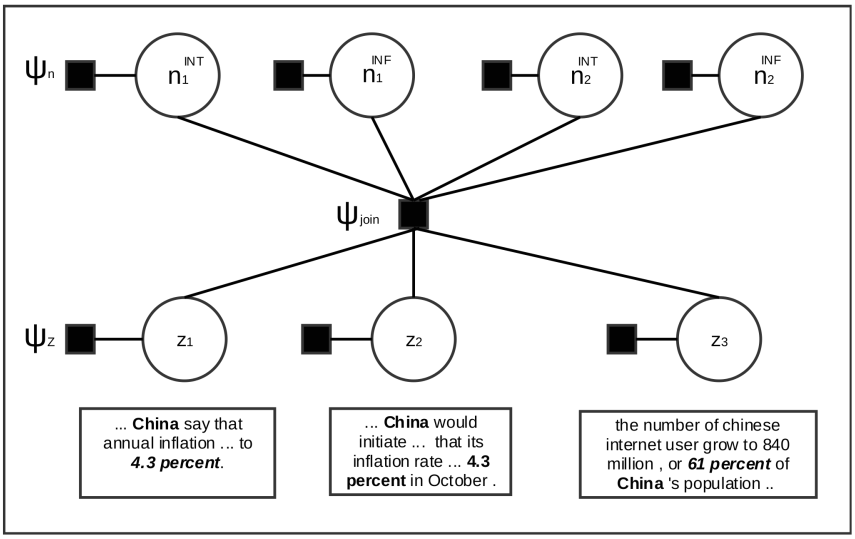

ψn
n1 INT
n1 INF
n2 INT
n2 INF
ψjoin
ψz
z1
z2
z3
... China say that annual inflation ... to 4.3 percent.
... China would initiate ... that its inflation rate ... 4.3 percent in October .
the number of chinese internet user grow to 840 million , or 61 percent of China 's population ..

Figure 6.1: The NumberTron Graphical Model for the Entity *China*

relation r. With each $s \in S_{e,q}$ is a multi-ary random variable z_s that can take values $r \in \mathcal{R} = (R \cup \perp)$ denoting if the sentence expresses any of the R relations or none of them (when $z_s = \perp$).

6.4 Potentials

The z variables are associated with node potentials: ψ_s, ψ_q^r derived respectively from a set of features: ϕ_s and ϕ_q and their associated parameters θ_s and θ_q^r. There are no node potentials associated with the n nodes. We describe the features in section 6.5. The **n** and **z** nodes are constrained by ψ^{join} potentials to ensure that **n** variables are one only under sufficient support from **z** variables and to incorporate agreement among close-by numbers. There are no parameters attached to these potentials. Thus, the joint distribution over labels of sentences that contain the entity e is

$$\begin{aligned}&\Pr(\mathbf{z}, \mathbf{n} | S_e, Q_e, \theta) = \\ &\frac{1}{\mathbb{Z}} \prod_{q \in Q_e} \prod_{r \in \mathcal{R}} \psi_q^r(n_q^r) \prod_{s \in S_{e,q}} \psi_s(z_s) \psi^{\text{join}}(\mathbf{n}, \mathbf{z})\end{aligned} \tag{6.1}$$

6.5 Features

We use three different kinds of features.

Feature type	Features
Fixed Keywords	key: life key: expect
All Keywords	key: life key: expect key: world
Number Features	Num: Billion Num: Integer

Table 6.1: Number and Keyword Features for the sentence *Afghanistan , which be mostly rural , have one of the lowest life expectancy rate in the world at 44 year for both man and woman.* We use the number and one of the features, along with the features described in (Mintz et al. 2009)

Mintz Features Proposed by (Mintz et al. 2009), these features are used by several state of the art distant supervision systems, sometimes with minor tweaks. We use these features as it is.

Keyword Features A set of words identified as important for identifying a relation are tagged as the keywords of the relation. We generate the keyword features in two different ways:

- Using the keyword database (Table A.1).
- By tagging all the nouns in the sentence as a potential keyword. The idea is that only the words that are actually indicative of the relation will be actually repeated and thus the weight distribution will weed out the noisy keywords. As discussed in results, there still are enough noisy keywords to cause trouble.

The keywords are always stemmed before adding.

Number Features Information on the magintude and type (whole, fraction) can also be useful for relation extraction. For example, population will never be a fraction, percent of internet users is likely to be a fraction and will always be a non-negative number less than equal to 100. We capture all these signals in the number features. The numbers are converted to the SI unit before adding using the unit tagger. For example, Life expectancy, expressed in years, is converted to seconds before adding.

We summarize the keyword and number features used in the table 6.1.

The time "44 year" is converted to the SI unit, which comes out to be around 1.3 billion and thus the feature Num: Billion is fired.

6.6 Parameter Learning

6.6.1 The Perceptron

Our training algorithm is based on the Perceptron (Rosenblatt 1958). The core idea of the (supervised) learning process is to update the weights iteratively so that the predictions made by the model align with the true labels. Given a set of weights w, the training set D consisting of a set of inputs $\{x_1, x_2, ..., x_N\}$. The weights are predicted as per the following rule:

$$w \leftarrow w + \eta * (t_i - o_i) * x_i \tag{6.2}$$

$\forall i \in D$. where o_i is the prediction $f(w.x_i)$ and t_i is the true label for the i^{th} example in the training data D. Our training algorithm learns the parameters θ of the model using distant supervision and a perceptron-like training algorithm Collins 2002.

6.6.2 Getting Supervision

We have two sources of information for getting true labels for the random variables in our model. The first is a database of numerical facts that we use to assign labels to the number nodes. We also have a list of keywords, 1-4 per relation, that can indicate the presence of a relation in the sentence.

- **True Labelings of n_q^r nodes** A number node n_q^r for an entity e with the value n_q is set to true if the knowledge base has a triple (e, r, v) with the value v such that $(1 - \delta_r) * v \leq n_q \leq (1 - \delta_r) * v$
- **True Labelings of z nodes** A $\bar{z}_s = r$ if it is connected to a $\bar{n}_e^r = 1$ and a keyword of r appears in sentence s, else it is $\perp$. We experiment with different configurations wherein the labeling for the z nodes depend on only the KB or only the keywords.

6.7 Training Algorithm

We use the Collins perceptron algorithm (Collins 2002) with regularization (K. Zhang, Su, and Zhou 2014) to train the θ parameters using the $\bar{\mathbf{n}}_e, \bar{\mathbf{z}}_e$ assignments over several entities e as labeled data. The training loop needs inference to find $\text{argmax}_{\mathbf{n},\,\mathbf{z}} \Pr(\mathbf{n}, \mathbf{z} | S_e, Q_e; \theta)$. We elaborate on how we solve this inference task.

Algorithm 3 The NumberTron Training Algorithm

```
A set of entities $\mathcal{E}$, for each $e \in \mathcal{E}$, the set of labelings $\bar{\mathbf{z}}_e, \bar{\mathbf{n}}_e$ from the gold-db, and the set of sentences $S_e$ and distinct quantities $Q_e$ that are connected to these labeled nodes.
initialize parameter vector $\theta \leftarrow 0$
for $t = 1 \ldots$ Training_iterations do
    for $e \in \mathcal{E}$ do
        $(\hat{\mathbf{n}}, \hat{\mathbf{z}}) \leftarrow \text{argmax}_{\mathbf{n},\mathbf{z}} \Pr(\mathbf{n}, \mathbf{z} | S_e, Q_e; \theta)$
        if $(\hat{\mathbf{n}}, \hat{\mathbf{z}}) \neq (\bar{\mathbf{n}}_e, \bar{\mathbf{z}}_e)$ then
            $\theta \leftarrow (1 - \alpha) * \theta + \phi(S_e, \bar{\mathbf{z}}_e, \bar{\mathbf{n}}_e) - \phi(S_e, \hat{\mathbf{z}}, \hat{\mathbf{n}})$
        end if
    end for
end for
```

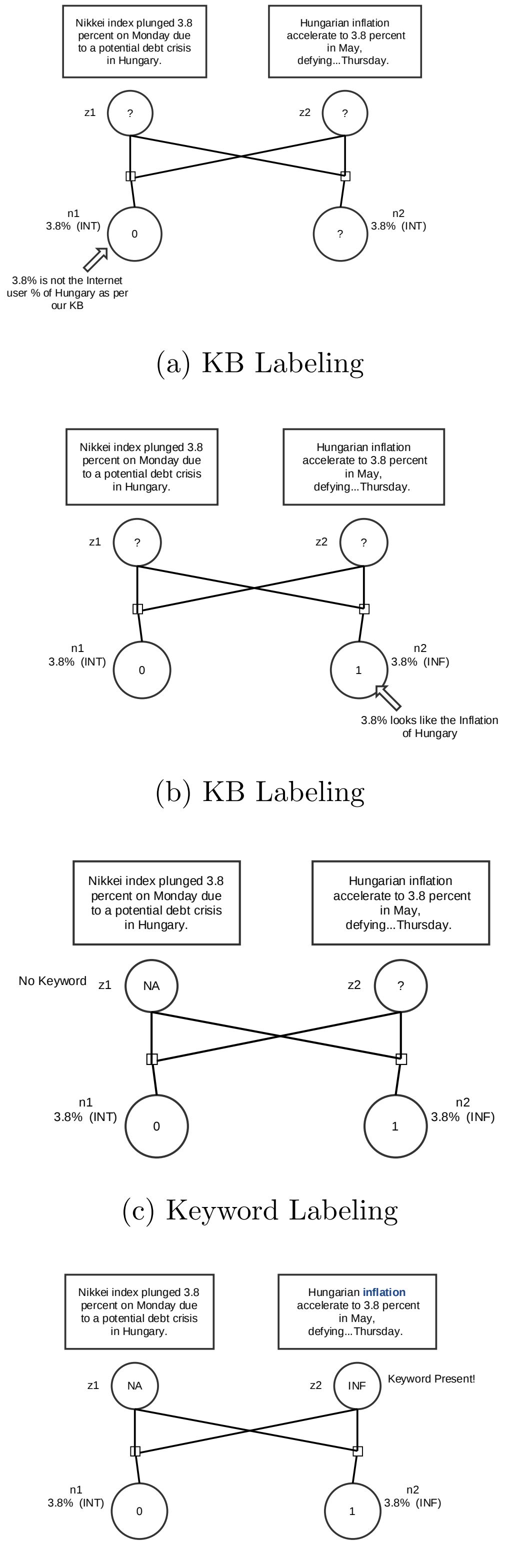
Inference Our goal was to design an efficient inference algorithm that can run in one pass over large training datasets. Accordingly, we first assign $\hat{z}_s = \text{argmax}_{r \in \mathcal{R}} \theta_s \phi_s(r)$ for each $s \in S_e$. We then assign the **n** variables based on the $\hat{z}_s$ variables and the constraints imposed by the ψ^{join} potentials. We experimented with the following definitions of the join potentials:

- **Simple OR**: n^r_q is set to one if and only if there exists any $s \in S_{e,q}$ such that $\hat{z}_s = r$.
- **Atleast-K**: n^r_q is set to one if and only if at least k fraction of $s \in S_{e,q}$ have $\hat{z}_s = r$. We use $k = 0.5$ for our experiments.
- **Agreeing-K**: For numbers, we want the one-labeled nodes to be proximal. In this scheme we start with the **Atleast-K** assignment $\hat{\mathbf{n}}$ and set to zero any n^r_q outside a range of $\pm\delta_r\%$ of a chosen central value. We choose the central value c for which $\hat{n}^r_c = 1$ and which causes smallest number of $\hat{n}^r_q = 1$ to set to zero.

The complete training process, getting supervision using the keyword database and the numerical fact database, and full inference, is explained with an example in figure 6.2 and figure 6.3.

6.8 Extraction

We perform sentence level extractions. Given a sentence S, let E be the set of entities and Q be the set of numbers that are present in the sentence. We then calculate a score(r, e, q) for a $e \in E$ and $q \in Q$ for being tagged r as $\theta^r_q \phi_q(n_q = 1) + \theta_s \phi_s(r)$ where ϕ_s captures the features in sentence S


Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
?
z2
?
n1
3.8% (INT)
0
n2
3.8% (INT)
?
3.8% is not the Internet user % of Hungary as per our KB

(a) KB Labeling

Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
?
z2
?
n1
3.8% (INT)
0
n2
3.8% (INF)
1
3.8% looks like the Inflation of Hungary

(b) KB Labeling

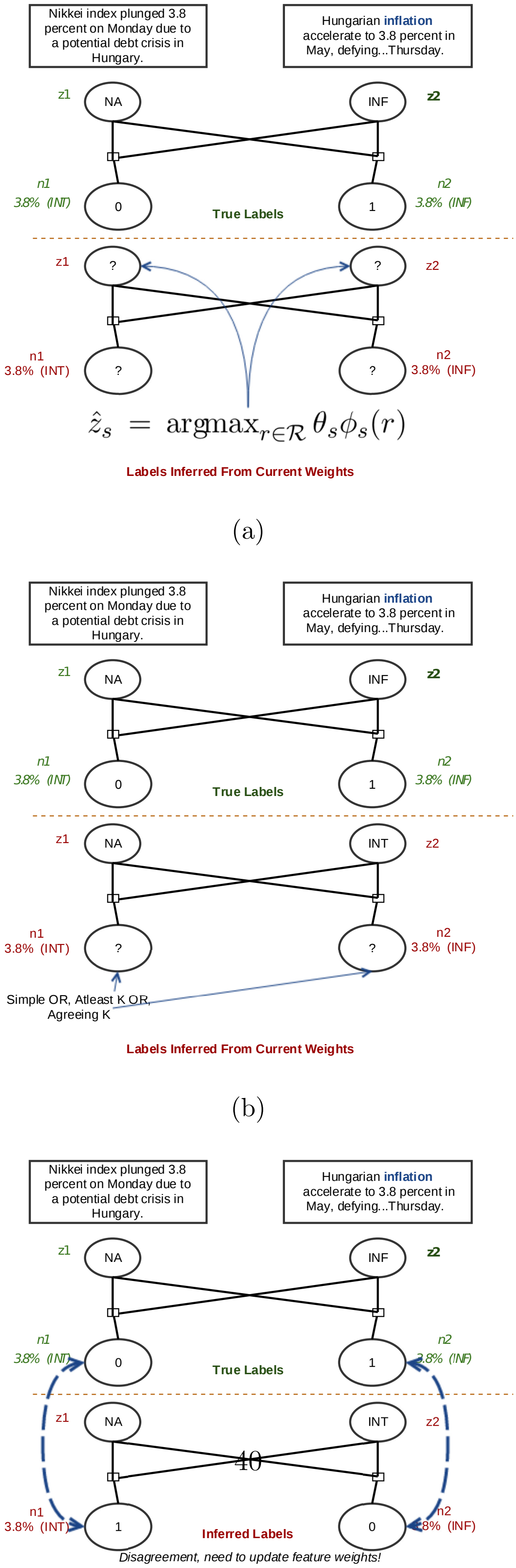
Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
No Keyword
z1
NA
z2
?
n1
3.8% (INT)
0
n2
3.8% (INF)
1

(c) Keyword Labeling

Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
NA
z2
INF
Keyword Present!
n1
3.8% (INT)
0
n2
3.8% (INF)
1

(d) Keyword Labeling

Figure 6.2: Obtaining True Labels


Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
NA
INF
z2
n1
3.8% (INT)
0
True Labels
1
n2
3.8% (INF)
z1
?
?
z2
n1
3.8% (INT)
?
?
n2
3.8% (INF)
$\hat{z}_s = \text{argmax}_{r\in\mathcal{R}} \theta_s \phi_s(r)$
Labels Inferred From Current Weights

(a)

Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
NA
INF
z2
n1
3.8% (INT)
0
True Labels
1
n2
3.8% (INF)
z1
NA
INT
z2
n1
3.8% (INT)
?
?
n2
3.8% (INF)
Simple OR, Atleast K OR, Agreeing K
Labels Inferred From Current Weights

(b)

Nikkei index plunged 3.8 percent on Monday due to a potential debt crisis in Hungary.
Hungarian inflation accelerate to 3.8 percent in May, defying...Thursday.
z1
NA
INF
z2
n1
3.8% (INT)
0
True Labels
1
n2
3.8% (INF)
z1
NA
INT
z2
n1
3.8% (INT)
1
Inferred Labels
0
n2
3.8% (INF)
Disagreement, need to update feature weights!

(c)

Figure 6.3: Full Inference: Obtaining Observed Labels

tied to entity e and number q. For each (e, q) we assign a label r if the min-max normalized score is greater than some threshold α. We use a cross validation set to obtain the $\alpha = 0.90$. The extraction procedure is presented as Algorithm 4

Algorithm 4 The NumberTron Extraction

```
Given a parameter vector, $\theta$ from a trained numbertron model
Given a sentence S, let $E$ be the set of entities and $N$ be the set of numbers that are present in the sentence.
Let $score(e, n)_r$ be the score of the entity-number pair (e, n) for a relation $r$.
for $(e, n) \in (E \times N)$ do //For all entity-number pairs
    for Relation $r$ in R do
        $score(e, n)_r \leftarrow \theta_r^m * \phi_s(S, r) + \theta_r^n * \phi_s(S, n) + \theta_r^m * \phi_s(S_i, r)$
    end for
    $score(e, n) \leftarrow min\_max\_normalization(score(e, n))$
    Let $r'$ be the relation with the highest normalized score
    if $score(e, n)_{r'} \geq \alpha$ then
        Extract $r'(e, n)$
    end if
end for
```

Chapter 7

Results and Discussions

This chapter elaborates on the test. We start with a discussion on the test bed: Corpus, KB and the test sentences. We present results, both aggregate and relation wise and provide analysis of the numbers with examples.

7.1 Testbed

7.1.1 Dataset

Corpus The training corpus that we are using is TAC KBP 2014 corpus. The corpus is pre-processed so that every sentence contain at least one numerical quantity. TAC KBP 2014 corpus consists around 3 million documents containing variety of sources like newswire, discussion forums, and web documents selected from GALE web collections.

Knowledge Base We compile our fact database by scraping data.worldbank.org.

This data from the world bank consists of numerical indicators about 249 countries. There are 1281 numerical attributes for these countries and a total of 4371979 facts. We perform experiments on selected indicators based on the availability of sentences which potentially express these facts in the corpus and to have some diversity in units.

7.1.2 Targeted Relations

For our experiments we picked up 10 relations from the 1281 numerical attributes available from the world bank dataset. These relations were selected based on the availability of training data, magnitude of the numerical quantity, similarities in the range of some of the numerical relations, and homogeneity of units.

INTERNET and INF are the relations with same unit and are overlapping in the range of the values and so is the case with GOODS, GDP, and FDI. POP is peculiar, because it has no unit and hence every number without a unit is possible candidate for POP which attracts unprecedented noise. In spirit of **soderlandtype** what NER is to traditional relation extraction, Units are to numerical relation extraction.

Relation	**Code**	**Unit**
Land Area	AGL	sq km
Annual Inflation	INF	percent
Internet Users	INTERNET	percent
Goods Export	GOODS	$ (USD)
GDP	GDP	$ (USD)
FDI	FDI	$ (USD)
CO_2 Emissions	CO_2	kiloton
Total Population	POP	
Life Expectancy	LIFE	year
Electricity Production	ELEC	kWh

Table 7.1: Relations used for experiments

7.1.3 Testset

The test corpus that we use consists of 430 sentences partially derived from the TAC corpus and sentences from the top web pages obtained from the search engine for the relations. Since, the training corpus contains a tremendous amount of noise, it was hard to sample out the true instances for testing. This is the reason for some of the true sentences to be derived from the web search.

7.1.4 Generating Spots for Testing

Given a test sentence, there are several ways in which the Number-Location pairs that are presented to the extractor(numbertron or numberule) for extraction are selected. One of the possible ways is to take the closest ones, and a third is to take a cross product and present all of them. For numerical relations, we expect that the number immediately following or preceding a country is the potential candidate. We call this scheme (***ProximalInstanceGeneration***) and the default scheme of generating all pairs ***DefaultInstanceGeneration***. We present results using only DefaultInstanceGeneration. This method is selected because the gain of precision while using ProximalInstanceGeneration was not worth the corresponding gain of recall.

Relation	Positive	Negative
AGL	57	17
POP	51	250
INF	51	84
INTERNET	15	
FDI	10	35
GDP	8	
GOODS	11	
LIFE	15	34
ELEC	13	6
CO_2	8	16

Table 7.2: Analysis of test data

For every number, based on the unit we see which of the relation it belongs to and if that number indeed express the relation, we say it positive for that relation, else we say negative for that relation class. The reason being, that the number was a valid candidate for all of the relation in relation class and it wasn't positive for any of them.

7.1.5 Adapting MultiR for Numerical Relation Extraction

To compare how we fare against the existing relation extractors, we tweaked a state of the art relation extraction system, MultiR Hoffmann, Congle Zhang, Ling, et al. 2011 for numerical relation extraction. We summarize the changes made to the MultiR code base obtained from `https://github.com/jgilme1/MultirExperiments`, commit 0b465a74dc49b298c.

- **Identifying Number as an argument** The numbers were identified as possible arguments apart from the usual named entities using a regular expression.
- **Flattening of the numbers using the unit tagger** The unit tagger was used to flatten the number to the SI units.
- **Modifying the relation matcher to perform matching from the fact database.** As discussed, we cannot expect the exact match with the numerical fact base to work. We thus tweaked the relation matching code in the MultiR to perform matching with some threshold. The

System	**Precision**	**Recall**	**F1 Score**
MultiR++	31.81	28.10	29.84
Recall-Prior	28.18	86.19	42.47
NumberRule	59.30	53.60	56.30
NumberTron	60.93	66.92	63.78

Table 7.3: Aggregate results. NumberTron outperforms all other methods.

thresholds used for MultiR and the amount of entries used were same for both MultiR and the NumberTron.

- **Unit Compatibility Check** We also equipped MultiR with the unit tagger to throw away extractions that had incompatible units. Without this capability, the numbers from MulitR were unbelievably bad.

We call this adaptation of MultiR **MultiR++**.

7.1.6 Recall-Prior Baseline

For a given unit, predicting the most common relation in the test set leads to this baseline. For example, for the relation percent, we predict inflation and so on. Since an extraction is always made, this baseline gives a perfect recall for the most frequent relation.

7.2 Results

7.2.1 Comparison of different methods

From table 7.3, it is evident that NumberRule performs well for numerical relations because of the way numerical facts are constructed. NumberTron which uses the same amount of information and training corpus performs at par with the NumberRule. These two systems are different in the sense that NumberTron can also extract the facts that are devoid of keywords, because of the syntactic and lexical patterns learn during the training. MultiR++ performs quite poorly. We believe this is because of different peculiarities of numerical relations that it doesn't handle.

7.2.2 NumberTron vs. NumberRule

Both the relatively simple rule based system discussed earlier and the much more sophasticated numertron rule based system cater to some niches of the

Distant Supervision	**Simple OR**			**Atleast-K**			**Agreeing-K**		
	P	R	F1	P	R	F1	P	R	F1
KB	43.24	50.93	46.54	40.05	53.93	45.97	35.20	44.52	39.35
Keywords	43.35	73.22	54.46	43.69	73.62	54.83	45.97	70.80	55.74
KB + Keywords	61.56	64.96	63.21	60.93	66.92	63.78	63.46	60.21	61.79

Table 7.4: Comparison of various configurations for NumberTron

Features	P	R	F1
Mintz features only	22.85	36.86	28.21
Keyword features only	51.24	52.55	51.89
Mintz + Keyword	47.10	39.04	42.71
Mintz + Number	17.80	35.03	23.67
Keyword + Number	45.15	69.70	54.80
Mintz + Key. + Num.	*60.93*	*66.92*	*63.78*

Table 7.5: Ablation tests of feature templates for NumberTron

Relation	NumTron F1	NumRule F1
FDI	0	50.00
Life Expectancy	68.96	69.50
Internet Users	55.73	54.54
Electricity Prod.	50.00	62.50
GDP	57.14	42.80
CO_2 Emissions	47.61	53.30
Inflation	88.40	56.25
Goods export	75.00	35.20
Population	49.99	60.30
Land Area	57.44	52.22

Table 7.6: Per relation F1 scores for NumberRule and best configuration of NumberTron

numerical relation family. We compare the extractions made by both the systems with the help of certain examples.

- *Turkey 's central bank say Wednesday it expect the annual inflation rate to reach 6.09 percent at the end of 2009 , lower than the official target of 7.5 percent .*
- **Extraction** Turkey 6.09 percent INF
- **NumberRule** fails to make an extraction because the keyword inflation is not on the extraction or this sentence, the keyword *inflation* is not on the dependency path shown below

  Turkey $\xrightarrow{poss}$ bank $\xrightarrow{nsubj}$ say $\xrightarrow{ccomp}$ expect $\xrightarrow{xcomp}$ reach $\xrightarrow{dobj}$ percent $\xrightarrow{num}$ 6.09
- **NumberTron** identifies the following features in the given sentence for the relation *inflation*
  - Num: Units (989944.1544936991)
  - `dep:[possessive]->|LOCATION|`
    `[poss]-> [nsubj]->[root] <-[ccomp]`
    `<-[xcomp]<- |PERCENT  (0.0)`
  - `dir:->|LOCATION|->-><-<-<-|PERCENT`
    (371801.1155436236)
  - `dir:LOCATION|->-><-<-<-|PERCENT|->`
    (1541308.1559610958)
  - `inverse_false|LOCATION|*LONG*`
    `|PERCENT`
    (1921578.015374796)
  - key: inflat (1.3685271144056765E7)
  - key: rate (-157991.38527419555)

Thus, apart from looking at the keyword, the NumberTron takes a hint from the type of number that is present (A number in units, that is, between 0 and 10), and the different dependency path patterns.

As another example, consider the sentence *Total annual co2 emissions grew from 1.25 billion tonnes in 1994 to 1.90 billion tonnes in 2007, confirming India among the world's biggest emitters.*

The extraction Co2 emission(India, 1.25 billion tonnes) is made by numberule but not numbertron. The former takes hint from the presence of a keyword, while the latter assigns a confidence of 0.448 to this extraction, because of a dependency path feature that has a large negative weight for the case when a location is attached to MONEY.

7.3 Analysis

We further analyze the strengths and weaknesses of NumberTron and NumberRule. NumberRule's missed recall is primarily because of not having a keyword on the dependency path. An illustrative example is: "*Turkey's central bank said Wednesday it expects the annual inflation rate to reach 6.09 percent at the end of 2009 , lower than the official target of 7.5 percent.*". From this sentence, NumberRule does not extract (Turkey, inflation rate, 6.09 percent), because the keyword 'inflation' is not on the shortest dependency path between Turkey and 6.09 (Turkey $\xrightarrow{poss}$ bank $\xrightarrow{nsubj}$ said $\xrightarrow{ccomp}$ expects $\xrightarrow{xcomp}$ reach $\xrightarrow{dobj}$ percent $\xrightarrow{num}$ 6.09). On the other hand, since NumberTron combines evidences from multiple features, it outputs this extraction. Several features such as number's range, presence of 'inflation' and 'rate' in the context and three different dependency path patterns fire for NumberTron.

Table 7.6 lists the F-scores of the two systems for each relation. By and large NumberTron wins on recall, and has performance within 10-15 points of NumberRule. However, for FDI relation, NumberTron does not output a single extraction! This is because sentences expressing this relation are rare in our training corpus.

On Goods and Population, NumberRule has an unusually weaker recall. Both these relations are well represented in the training corpus making it easier for NumberTron to learn. Moreover, NumberRule's test 4 significantly reduces recall for these – many sentences in our testset mention multiple values for the same entity-relation in a sentence, from which NumberRule extracts only the first. An (abridged) example is "*Annual average inflation for Lithuania fell to 7.9 percent in July from 8.7 percent in June and 9.4 percent in May.*".

Finally, population relation is unusual in that NumberRule has high recall and low precision, and NumberTron is exactly reverse. This was because one of the pre-keywords was 'people'. This is a generic word and led to many errors for NumberRule. On the other hand, NumberTron powered by the KB learns low weight for this keyword, and improves precision, but this also hurts recall.

7.4 Ablation Study for NumberTron

We now report the experiments that help us in identifying the best configurations for NumberTron. In Section 6.6 we describe three choices for the design of ψ^{join} potential – Simple OR, Atleast-K, and Agreeing-K. Moreover, we implemented three different approaches for labeling the training data ($\bar{\mathbf{z}}_e$ variables) – (1) heuristically label all sentences with the right unit, keyword and entity as positive label, (2) distant supervision using KB, and (3) both keyword-based and KB-based distant supervision. This results in nine different configurations. Table 7.4 presents a comparison.

We verify from this experiment that standard distant supervision offers very weak signal for numeric extraction – results on KB only are not very good. Keywords are crucial, and KB in conjunction with keyword-based labeling adds significant value. We also learn that Atleast-K provides marginally better results than Simple OR. The Agreeing-K potential that enforces numbers to be within a band of δ is not as good, possibly because in the early stages of training, when the parameters are not well-trained, this is too severe a restriction. Overall we select Atleast-K in conjunction with KB + Keywords-based labeling as the best setting.

We also study the impact of the various features in node potentials of NumberTron. These include the original Mintz features Mintz et al. 2009, keyword-based features, and various number-specific features as discussed in Section 6.1. Table 7.5 presents the results. We find that by themselves the large set of Mintz features confuses the classifier; keyword features are much more effective. Number features substantially improve F1 in the presence of keywords. Combining all three yields the best performance.

Chapter 8

Summary and Conclusions

In chapter 4, we discussed several challenges that make numerical relation extraction a much more harder problem. Our rule based system NumberRule and the probabilistic system NumberTron both use novel techniques for addressing these challenges. Table 8.1 presents the approaches used by both these systems along with a comparison. We next list a number of future improvements that can further improve the performance of our system:

- **Temporal Modeling** Many of the relations that we target are time dependent. At the moment, there is no provision of inculcating time during training or extraction. At the moment, we extract patterns that possibly express one or more relations of interest. However, populating knowledge bases with reliable values of these relations would involve adding the notion of time. There are broadly two ways in which time specific information can be inculcated in the pipeline.
  - *Matching* Our knowledge base has facts on a relation from different times (inflation of India in 2006, for example), thus time cognizant matching may drastically reduce the number of false postives.
  - *Extraction* We can look for mention of dates during extraction to associate a fact with a time. This method can be expected to work because the expressions that express a particular relation all can be expected to have a similar structure, and time associated with the fact can be extracted by using a set of rules.

  Clearly, adding time during matching or extraction cannot solely rely on the presence of time in the mention, having information on the date on which the document was published can be of help.
- **Soft constraints in NumberTron** Instead of hardcoding true assignments to the random variables, we can think of an alternative scheme in

which the keyword nodes are added to the graphical model along with edge potentials that capture the similarity between potential relation and the keywords. We can also possibly add features that can capture insights like coherency in the keywords.

- *Extracting with Delta words* NumberTron and NumberRule ignore mentions that express a change rather than the absolute fact. It might be interesting for some numerical relations (like share price) to identify and extract such change expressing facts.

We present the first detailed study of the task of numerical relation extraction, in which one of the arguments of the relation is a quantity. Our preliminary analysis reveals several peculiarities that make the task differently challenging from standard IE. We employ these insights into a rule-based system, NumberRule, that can extract any numerical relation given input keywords for that relation. We also develop a probabilistic graphical model, NumberTron, that employs novel task-specific features and can be trained via distant supervision or other heuristic labelings.

By aggregating evidence from multiple features, NumberTron produces much higher recall at comparable precision compared to NumberRule. Both systems vastly outperform baselines and non-numeric IE systems, with NumberTron yielding over 20 point F-score improvement over the baseline and 33 F-score improvement over MultiR.

	NumberRule	**NumberTron**
Idea	Dependency path between the number and the entity in the mention is used to look for hints that may lead to extraction (keywords, extractions, dependency types).	A Graphical Model with Perceptron like training algorithm, True labels obtained using a numerical fact table and keyword list.
Supervision	List of relation specific keywords.	List of relation specific keywords, a numerical knowledge base.
Handling False Positives	Look for relation specific keywords in the dependency path.	Keyword features.
Handling Mentions Expressing Change	No extraction if a delta word exists on the dependency path.	Remove sentences having delta words on the dependency path during training and extraction.
Use of Unit Tagger	Units are used to test compatibility of a relation and the number.	Units are used for training data creation and flattening to SI units.
Common Number Pruning	N/A	Features included to capture type (whole, fraction), magnitude and frequency.
Modified Relations	Handled by forming relation phrases: attaching keywords with words related via modifying dependencies, like *urban* population.	Not handled in the model, can be handled at the time of extraction using a scheme similar to the one used by NumerRule.
Results	P = 59.30, R = 53.60, F-Score = 56.30	P = 60.93, R = 66.92, F-Score = 63.78

Table 8.1: NumberTron and NumberRule

Appendix A

Appendix

A.1 Likelihood expression

For a given location-relation pair
$O(\theta) = P(DB|S, N; \theta)$
$= \sum_{n,z} P(DB, n, z|S, N; \theta)$

$= \sum_{n,z} \frac{1}{\mathbb{Z}} * \prod_{i=1}^{N_S} \psi_s(S_i, z_i) * \prod_{k=1}^{N_n} \psi_n(N_k, n_k) * \prod_{P,Q \in T} \psi_{nn}(P, Q)$

Where T is the set of (n, DB) nodes that are connected.only because you'll get to appreciate how good of a teacher he is.

$= \frac{1}{\mathbb{Z}} * \sum_{n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i, z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k, n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P, Q))$

Which gives

$log(O(\theta)) = log(\sum_{n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i, z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k, n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P, Q))) - log(\mathbb{Z})$
where
$\mathbb{Z} = \sum_{DB,n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i, z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k, n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P, Q))$

A.1.1 Gradient

$\frac{\partial(log(O(\theta)))}{\partial(\theta_j^s)} =$
$\frac{\sum_{n,z} {\phi_s}^j(S,z) * \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i,z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k,n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P,Q))}{\sum_{n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i,z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k,n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P,Q))}$
- $\frac{\sum_{DB,n,z} {\phi_s}^j(S,z) * \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i,z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k,n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P,Q))}{\sum_{DB,n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i,z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k,n_k) + \sum_{P,Q \in T} \theta^{nn} \phi_{nn}(P,Q))}$

$= E_{P(n,z|DB,S,N;\theta)}[{\phi_s}^j(S,z)]$ - $E_{P(n,z,DB|S,N;\theta)}[{\phi_s}^j(S,z)]$

Where ${\phi_s}^j(S,z) = \sum_{i=1}^{N_S} {\phi_s}^j(S_i,z_i)$

As can be seen from the expression of likelihood, similar expressions of gradients are obtained for all the θs.

The gradient expression is identical in structure to the expression of the multiR model, and hence we can think of using similar schemes for inference and training. The only thing missing is the deterministic OR potentials, it is not immediately clear how would they fit into the model.

A.2 Resources

We have released our code at `http://www.github.com/neo-ie`. Specifically,

- `https://github.com/NEO-IE/NumberRule` is the rule based system discussed in chapter 5.
- `https://github.com/NEO-IE/numbertron` is the initial implementation of the NumberTron B.
- `https://github.com/NEO-IE/numbertron/tree/pstod` is an alternative implementation of the NumberTron with nary z nodes.
- `https://github.com/NEO-IE/MultirExperiments` is our tweaked version of MultiR.
- `https://github.com/NEO-IE/Hadoop-Scripts` is the set of hadoop scripts that we used to preprocess the corpus. In particular, `https://github.com/NEO-IE/Hadoop-Scripts/blob/master/pipeline.sh` is the script that orchestrates the entire NLP pipeline.
- `https://github.com/NEO-IE/numrelkb/blob/master/keywords_small.json` the set of keywords used for different relations.
- `https://github.com/NEO-IE/numrelkb/blob/master/kb-worldbank-SI.tsv` the numerical fact knowledge base derived from `data.worldbank.org` and standardized to the SI units.

A.3 List of Keywords

Relation	Keywords
Internet User %	internet
Land Area	area, land
Population	population, people, inhabitants
GDP	gross, domestic, GDP
CO_2 emission	carbon, emission, CO2, kilotons
Inflation	inflation
FDI	foreign, direct, investment, FDI
Goods Export	goods, export
Life Expectancy	life, expectancy
Electricity Production	electricity

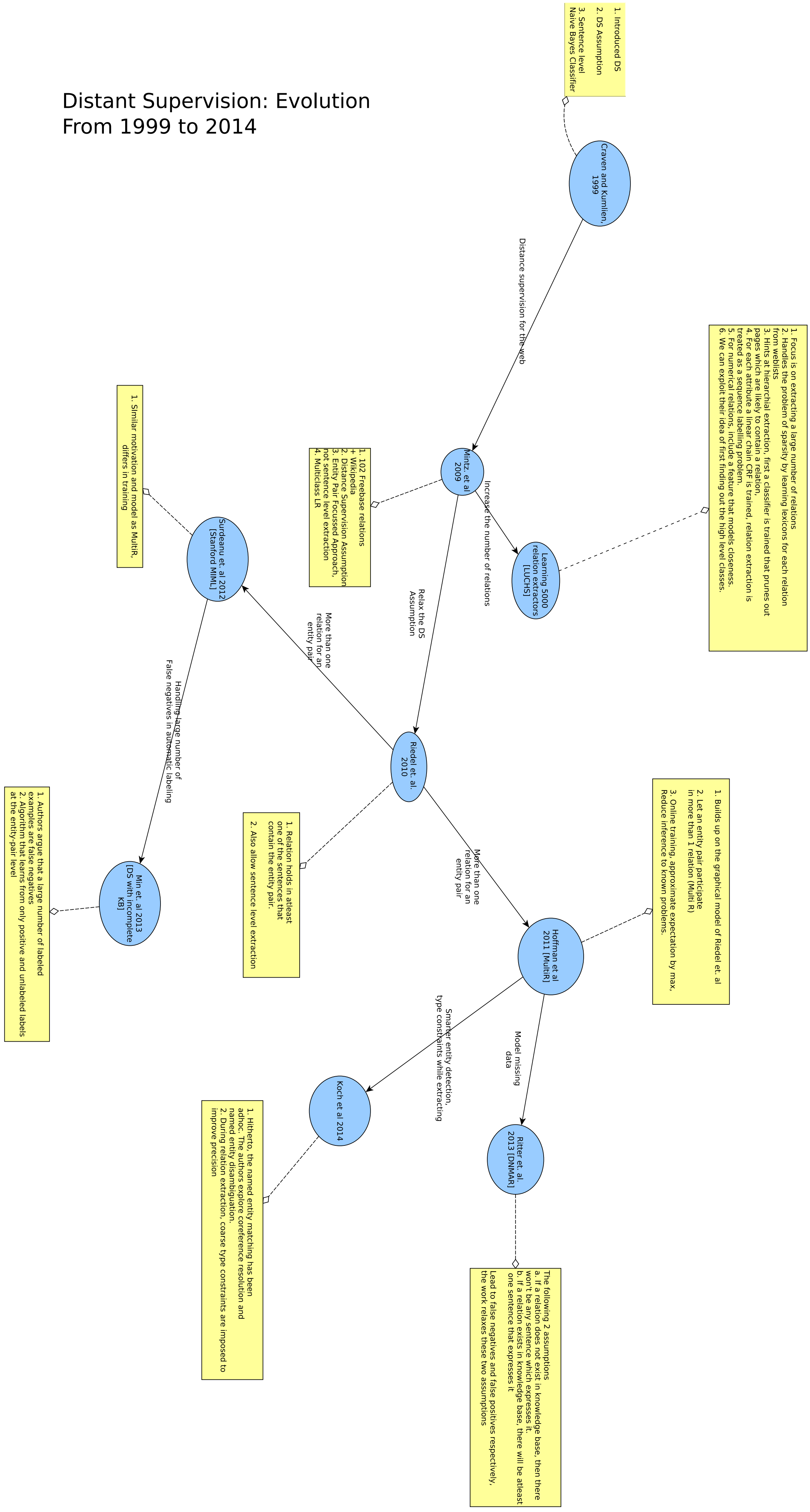
Table A.1: Pre-specified keywords

A.4 List of Delta words

change, up, down, growth, increase
decrease, decreased, increased, changed
grown, grew, surge, surged, rose, risen

Table A.2: The set of delta words

Distant Supervision: Evolution From 1999 to 2014



Craven and Kumlien, 1999

1. Introduced DS
2. DS Assumption
3. Sentence level Naive Bayes Classifier

Distance supervision for the web

Mintz. et al 2009

1. 102 Freebase relations + Wikipedia
2. Distance Supervision Assumption
3. Entity Pair Focussed Approach, not sentence level extraction
4. Multiclass LR

Increase the number of relations

Learning 5000 relation extractors [LUCHS]

1. Focus is on extracting a large number of relations
2. Handles the problem of sparsity by learning lexicons for each relation from weblists
3. Hints at hierarchial extraction, first a classifier is trained that prunes out pages which are likely to contain a relation.
4. For each attribute a linear chain CRF is trained, relation extraction is treated as a sequence labelling problem.
5. For numerical relations, include a feature that models closeness.
6. We can exploit their idea of first finding out the high level classes.

Relax the DS Assumption

Riedel et. al. 2010

1. Relation holds in atleast one of the sentences that contain the entity pair.
2. Also allow sentence level extraction

More than one relation for an entity pair

Surdeanu et. al 2012 [Stanford MIML]

1. Similar motivation and model as MultiR, differs in training

Handling large number of False negatives in automatic labeling

Min et. al 2013 [DS with incomplete KB]

1. Authors argue that a large number of labeled examples are false negatives
2. Algorithm that learns from only positive and unlabeled labels at the entity-pair level

More than one relation for an entity pair

Hoffman et al 2011 [MultiR]

1. Builds up on the graphical model of Riedel et. al
2. Let an entity pair participate in more than 1 relation (Multi R)
3. Online training, approximate expectation by max, Reduce inference to known problems.

Smarter entity detection, type constraints while extracting

Koch et al 2014

1. Hitherto, the named entity matching has been adhoc. The authors explore coreference resolution and named entity disambiguation.
2. During relation extraction, coarse type constraints are imposed to improve precision

Model missing data

Ritter et. al. 2013 [DNMAR]

The following 2 assumptions
a. If a relation does not exist in knowledge base, then there won't be any sentence which expresses it.
b. If a relation exists in knowledge base, there will be atleast one sentence that expresses it
Lead to false negatives and false positives respectively, the work relaxes these two assumptions

Appendix B

NumberTron with binary z nodes

B.1 Introduction

We discuss an earlier version of the numbertron that had binary z nodes.

B.2 The Graphical Model

There is one graph for every entity-relation pair, called the *entity-relation graph.*

An example graphical model for the location-relation pair 'Afghanistan-Life Expectancy' is illustrated in Figure B.1. We assume that all such sentences pertain to the location of interest ('Afghanistan', in this case) and have mention of some number of the same type as the relation of interest ('year' in this case). The graph is over two types of random variables and correspondingly has two types of nodes:

1. node corresponding to a sentence of potential match (*e.g.*, $z1$ and $z2$ in Figure B.1). Each such node corresponds to a binary random variable indicating the relevance of the sentence to the location-relation pair of interest. There is a local factor for each such node, which provides some evidence for the relevance of the sentence to the relation. This local factor is a linear combination of features that include

    - standard distant supervision systems like MultiR use lexical and syntactic features adopted from (Mintz et al. 2009)
    - the presence of some representative keyword which we will hereafter refer to as the *keyword feature.*

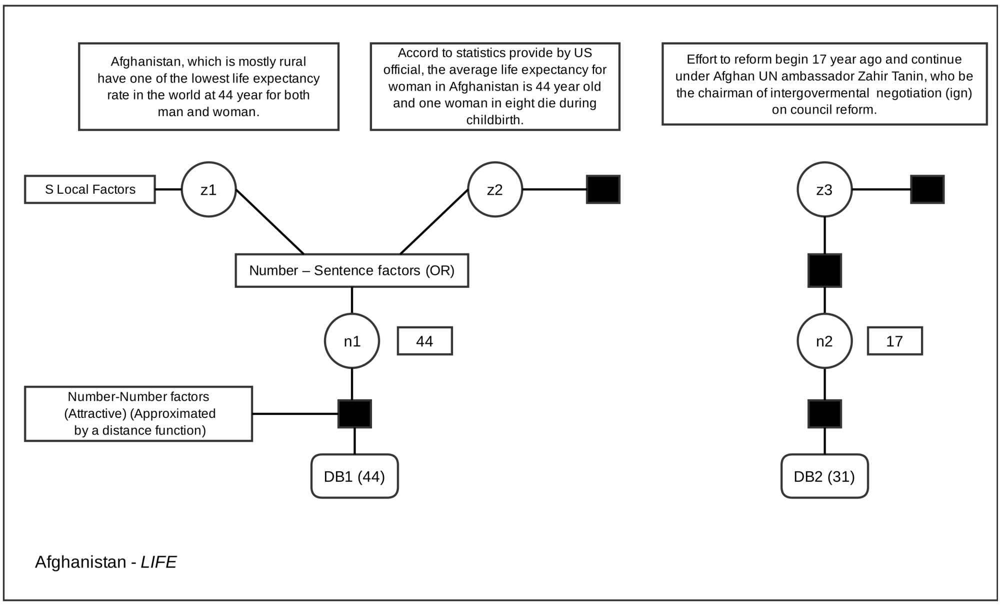

Afghanistan, which is mostly rural have one of the lowest life expectancy rate in the world at 44 year for both man and woman.
Accord to statistics provide by US official, the average life expectancy for woman in Afghanistan is 44 year old and one woman in eight die during childbirth.
Effort to reform begin 17 year ago and continue under Afghan UN ambassador Zahir Tanin, who be the chairman of intergovermental negotiation (ign) on council reform.
S Local Factors
z1
z2
z3
Number – Sentence factors (OR)
n1
44
n2
17
Number-Number factors (Attractive) (Approximated by a distance function)
DB1 (44)
DB2 (31)
Afghanistan - *LIFE*

Figure B.1: A Sample Location-Relation Graph for Afghanistan-Life Expectancy

These features have also been incorporated in NumberTron.

2. node corresponding to a canonical number occurring across sentences (*e.g.*, $n1$) . Again, each such node corresponds to a binary random variable indicating whether the number is an instance of the relation for the country of interest. There is also a local factor for each such node, in the form of linear combination of features such as whether the number is a small whole number, the rarity (idf) of such a number in the corpus and so on. Refer to Section 6.5 for further description of these features.

B.2.1 Features

The features are the same as described in Chapter 6.

B.3 Algorithm: Learning and Inference

The parameters of the graphical model are shared across all locations and all relations that have the same unit as the given relation. These parameters are learnt in a semi-supervised manner using the distant supervision approach.

Succinctly, the numbertron is based on the perceptron as described in (Collins 2002). The idea is to obtain a configuration of the **n** and **z** nodes given a configuration of the parameters, and then compare this with the configuration on the **z** and **n** nodes obtained by using the gold database. We then update the parameter settings depending on whether or not these configurations agree.

B.3.1 DB Nodes

The DB nodes store the fact database that is used for supervision during the training. These can be considered to be the real valued nodes that are connected to the number node that has the closest corresponding magnitude.

- Learning Algorithm and Inference
- Explain Fullinference and Conditional inference
- Discuss the variations in the Conditional Inference
  - GoldDb Inference
  - Keyword Inference
  - GoldDb + keyword Inference

Learning Objective We now formally define the learning objective and draw parallel with MultiR. We note that although the expressions look deceptively same, our location relation graph makes both conditional and full inference relatively straight forward, as opposed to MultiR, wherein the conditional inference was tricky.

We define the likelihood of a given location relation graph as follows:

For a given location-relation pair

$$\begin{aligned} O(\theta) &= P(DB|S,N;\theta) \\ &= \sum_{n,z} P(DB,n,z|S,N;\theta) \\ &= \sum_{n,z} \frac{1}{\mathbb{Z}} * \prod_{i=1}^{N_S} \psi_s(S_i,z_i) * \prod_{k=1}^{N_n} \psi_n(N_k,n_k) \end{aligned} \tag{B.1}$$

Where

$$\mathbb{Z} = \sum_{DB,n,z} \exp(\sum_{i=1}^{N_S} \theta^s \phi_s(S_i,z_i) + \sum_{k=1}^{N_n} \theta^n \phi_n(N_k,n_k) \tag{B.2}$$

and ϕ_s and ϕ_n are the sentence and the number features respectively as described in section 6.5.

The idea is to update the weights with the gradient whenever an update is required. We thus next present the expression for the gradient of the objective:

$$\frac{\partial(log(O(\theta)))}{\partial(\theta^s_j)} = \\ E_{P(n,z|DB,S,N;\theta)}[{\phi_s}^j(S,z)] - E_{P(n,z,DB|S,N;\theta}[{\phi_s}^j(S,z)] \tag{B.3}$$

Algorithm 5 shows the training algorithm psuedocode

B.3.2 Psuedocode

Algorithm 5 The NumberTron Training Algorithm

The training set is $\{(S_i, N_i)|i = 1...|L|\}$, i corresponds to a particular entity-relation pair, S_i consists of all the sentences corresponding to a particular entity relation and N_i corresponds to all the numbers found in these instances. T is the number of iterations, L is the total number of entity relation pairs in the training set. DB_i represents the database entries for the given relation.

```
initialize parameter vector $\theta \leftarrow 0$
for $t = 1...T$ do
    for $i = 1...L$ do
        $(\hat{\mathbf{n}}, \hat{\mathbf{z}}) \leftarrow \text{argmax}_{\mathbf{n},\mathbf{z},\mathbf{DB}}\, p(\mathbf{n}, \mathbf{z}, \mathbf{DB}|S_i, N_i; \theta)$
        $(\mathbf{n}^*, \mathbf{z}^*) \leftarrow \text{argmax}_{\mathbf{n},\mathbf{z}}\, p(\mathbf{n}, \mathbf{z}|DB, S_i, N_i; \theta)$
        if $\hat{\mathbf{n}} \neq \mathbf{n}_*$ then
            $\theta \leftarrow \theta + \phi(S_i, \mathbf{z}^*, \mathbf{n}^*) - \phi(S_i, \hat{\mathbf{z}}, \hat{\mathbf{n}})$
        end if
    end for
end for
```

We also approximate the expected values with max as in the case of MultiR. However, since no Z node is attached to more than 1 n node, the conditional Inference becomes easier. We note that the algorithm requires calculation of 2 expectations. We approximate both these expectations with maximization, as done by (Hoffmann, Congle Zhang, Ling, et al. 2011). We next discuss the details on Full inference and Conditional inference.

B.4 The *loose* closed world assumption

The *db*, or the database nodes capture our knowledge about the world. There are several ways of passing this knowledge to the training algorithm or equivalently, from the db nodes to the n nodes. One of the ways is by using an attractive potential function on the db node and number node edge. One could make such potentials extreme and enforce exact equivalence in the number present in the number node and the one obtained from the database. However, sometimes there are discrepancies in reporting of numerical facts across the sources. Moreover, some numerical quantities like the percentage of Internet users keep on changing with time and hence, there is only so much one can trust the gold db.

We thus set an n node to true if the database node *db* closest to it is within some range, say ± $\delta\%$ of the original value. This further simplifies our training procedure, since state of number nodes become known from the database.

B.5 Full inference, calculating $\operatorname{argmax}_{\mathbf{n},\ \mathbf{z},\mathbf{DB}}\, p(\mathbf{n}, \mathbf{z}, \mathbf{DB}|S_i, N_i;$

From the discussions in section B.4, the actual full inference expression becomes $\operatorname{argmax}_{\mathbf{n},\ \mathbf{z}}\, p(\mathbf{n}, \mathbf{z}, |S_i, N_i, DB; \theta)$

We can thus flip the z nodes first and then flip the n nodes depending on the configuration of the z nodes.

A z_i in the graph for location l_i and relation r_i is set to 1 if:
$\theta_{r_i} * \phi_s^r(S_i, z_i) >= \theta_{r'} * \phi_s^r(S_i, z_i) \forall r' \in r$

This is to say, that the weighted sum of the features fired for z_i should be the highest for the relation r_i.

Upon obtaining the best possible flips for the z nodes, the n nodes are flipped by taking an OR of the corresponding z nodes.

- **Simple OR** We set the number nodes n connected to the mention nodes z's to one if one of the z nodes is 1.
- **Atleast-K OR** We set the number nodes n connected to the mention nodes z to one if atleast a fraction k of the connected mention nodes are set to 1. We use $k = 0.5$ for our experiments.

Full inference involves one pass over the instance nodes. At each instance node, we need to calculate the score of that instance for all relations. the complexity of Full Inference thus is $\mathcal{O}((|S|*|R|)$ where $|S|$ is the total number of instances, and $|R|$ is the total number of nodes.

B.6 Conditional inference, calculating $\operatorname{argmax}_{\mathbf{n},\ \mathbf{z}}\, p(\mathbf{n},\ \mathbf{z}|DB,$

Again from the discussion in section B.4, the conditional inference problem becomes $\operatorname{argmax}_{\mathbf{z}}\, p(\mathbf{z}|db, \mathbf{n}, S_i, N_i; \theta)$

Moreover, each z is connected to exactly one n node. So the conditional inference becomes easy. We now define several conditional inference schemes that we have tried.

B.6.1 GoldDB Inference

We get the state of n nodes from db nodes using a distance function $\pm\delta\%$. For each n node that is false, we set all the corresponding z to be false. For each n node that is true, we set all the corresponding z to be true.

This is simple, and more importantly, trusting the gold database too much. As seen in section 4, there are several distinctions that set numerical relations apart from the usual entity-entity relations. One of the most important of these differences is the fact that only numbers may attract lots of noise. Though we reduce it to a great extent by pruning on units during graph creation, there still may be considerable noise that is attracted by only considering the numbers. We also noted that usually, an instance actually expressing a relation will have the relation name or a semantically close keyword. We modify our conditional inference procedure to capture these insights as defined in the subsequent sections.

B.6.2 Keyword Inference

(Pershina et al. 2014) learned several high quality features from a cleaner dataset, and these features were given a priority over the other features during training. On similar lines, for a location-relation graph for location l and relation r, we flip a z node to 1 if the instanec contains one of the keywords for the relation r. The n nodes are then flipped by using the z nodes.

B.6.3 GoldDB + Keyword Inference

This is the an obvious follow up of the conditional inference methods discussed above. In this method, the z nodes are first flipped using the GoldDB inference. Subsequently, the state of the individual z nodes are flipped by checking for keywords as in the keyword inference.

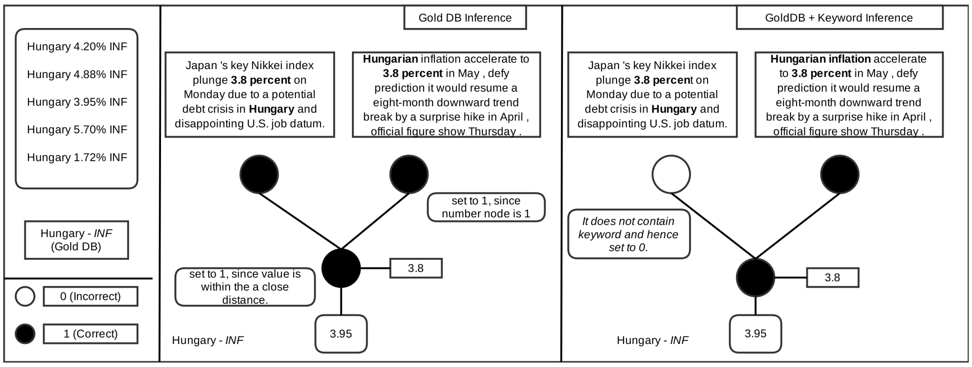

Hungary 4.20% INF
Hungary 4.88% INF
Hungary 3.95% INF
Hungary 5.70% INF
Hungary 1.72% INF
Hungary - *INF* (Gold DB)
0 (Incorrect)
1 (Correct)
Gold DB Inference
Japan 's key Nikkei index plunge **3.8 percent** on Monday due to a potential debt crisis in **Hungary** and disappointing U.S. job datum.
Hungarian inflation accelerate to **3.8 percent** in May , defy prediction it would resume a eight-month downward trend break by a surprise hike in April , official figure show Thursday .
set to 1, since number node is 1
3.8
set to 1, since value is within the a close distance.
Hungary - *INF*
3.95
GoldDB + Keyword Inference
Japan 's key Nikkei index plunge **3.8 percent** on Monday due to a potential debt crisis in **Hungary** and disappointing U.S. job datum.
Hungarian inflation accelerate to **3.8 percent** in May , defy prediction it would resume a eight-month downward trend break by a surprise hike in April , official figure show Thursday .
It does not contain keyword and hence set to 0.
3.8
Hungary - *INF*
3.95

Figure B.2: GoldDB and GoldDB + Keyword Inference

B.6.4 Active Inference

This is similar to GoldDB + keyword inference method, but the labels for the instance nodes are obtained from an oracle. Since the number of instance nodes is very large

All of these inferences involve doing one pass over the number nodes and one pass over the instance nodes, for a total complexity of The complexity of Full Inference thus is $\mathcal{O}((|S|+|N|)$ where $|S|, |N|$ are the total number of instances and the number nodes respectively.

B.7 Extraction

We perform sentence level extractions. Given an instance I, we tag it with a relation r if the min-max normalized score of the instance according to the given feature weights is greater than some threshold α. We use a cross validation set to obtain the $\alpha = 0.90$ The extraction procedure is same as presented as Algorithm 4.

Bibliography

[AG00] Eugene Agichtein and Luis Gravano. "*Snowball*: extracting relations from large plain-text collections". In: *ACM DL*. 2000, pp. 85–94.

[AZ11] Asma Ben Abacha and Pierre Zweigenbaum. "Automatic extraction of semantic relations between medical entities: a rule based approach". In: *J. Biomedical Semantics* 2.S-5 (2011), S4. URL: `http://www.jbiomedsem.com/content/2/S5/S4`.

[BM05] Razvan C. Bunescu and Raymond J. Mooney. "A Shortest Path Dependency Kernel for Relation Extraction". In: *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. 2005.

[CK99] Mark Craven and Johan Kumlien. "Constructing Biological Knowledge Bases by Extracting Information from Text Sources". In: *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, August 6-10, 1999, Heidelberg, Germany*. 1999, pp. 77–86.

[CLR13] Laura Chiticariu, Yunyao Li, and Frederick R Reiss. "Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!" In: *EMNLP*. 2013, pp. 827–832.

[Col02] Michael Collins. "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms". In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. EMNLP '02. 2002, pp. 1–8.

[DR10] Dmitry Davidov and Ari Rappoport. "Extraction and Approximation of Numerical Attributes from the Web". In: *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for*

Computational Linguistics, July 11-16, 2010, Uppsala, Sweden. 2010, pp. 1308–1317. URL: http://www.aclweb.org/anthology/P10-1133.

[FKZ07] Katrin Fundel, Robert Küffner, and Ralf Zimmer. "RelEx - Relation extraction using dependency parse trees". In: *Bioinformatics* 23.3 (2007), pp. 365–371. DOI: 10.1093/bioinformatics/btl616. URL: http://dx.doi.org/10.1093/bioinformatics/btl616.

[Gra14] Edouard Grave. "A convex relaxation for weakly supervised relation extraction". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.* 2014, pp. 1580–1590. URL: http://aclweb.org/anthology/D/D14/D14-1166.pdf.

[Hea92] Marti A. Hearst. "Automatic Acquisition of Hyponyms from Large Text Corpora". In: *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992.* 1992, pp. 539–545. URL: http://aclweb.org/anthology/C92-2082.

[Hof+11] Raphael Hoffmann, Congle Zhang, Xiao Ling, et al. "Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA.* 2011, pp. 541–550.

[HZW10] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. "Learning 5000 Relational Extractors". In: *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden.* 2010, pp. 286–295.

[Koc+14] Mitchell Koch et al. "Type-Aware Distantly Supervised Relation Extraction with Linked Arguments". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.* 2014, pp. 1891–1901.

[Man+14] Christopher D. Manning et al. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. 2014, pp. 55–60.

[Min+09] Mike Mintz et al. "Distant supervision for relation extraction without labeled data". In: *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. 2009, pp. 1003–1011.

[Per+14] Maria Pershina et al. "Infusion of Labeled Data into Distant Supervision for Relation Extraction". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*. 2014, pp. 732–738.

[Rit+13] Alan Ritter et al. "Modeling Missing Data in Distant Supervision for Information Extraction". In: *TACL* 1 (2013), pp. 367–378.

[Ros58] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[RVR15] Subhro Roy, Tim Vieira, and Dan Roth. "Reasoning about Quantities in Natural Language". In: *TACL* 3 (2015), pp. 1–13.

[RYM10] Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling Relations and Their Mentions without Labeled Text". In: *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*. 2010, pp. 148–163.

[Sar08] Sunita Sarawagi. "Information Extraction". In: *Foundations and Trends in Databases* 1.3 (2008), pp. 261–377. DOI: 10.1561/1900000003. URL: http://dx.doi.org/10.1561/1900000003.

[SC14] Sunita Sarawagi and Soumen Chakrabarti. "Open-domain quantity queries on web tables: annotation, response, and consensus models". In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. 2014, pp. 711–720. DOI: 10.1145/2623330.2623749. URL: http://doi.acm.org/10.1145/2623330.2623749.

[Sur+12] Mihai Surdeanu et al. “Multi-instance Multi-label Learning for Relation Extraction”. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea.* 2012, pp. 455–465.

[Zha+09] Chunju Zhang et al. “Rule-based extraction of spatial relations in natural language text”. In: *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on.* IEEE. 2009, pp. 1–4.

[ZSZ14] Kaixu Zhang, Jinsong Su, and Changle Zhou. “Regularized Structured Perceptron: A Case Study on Chinese Word Segmentation, POS Tagging and Parsing”. In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden.* 2014, pp. 164–173. URL: `http://aclweb.org/anthology/E/E14/E14-1018.pdf`.