DISTANT SUPERVISION USING PROBABILISTIC GRAPHICAL MODELS

Presented by:

Sankalan Pal Chowdhury

HUMAN SUPERVISION

Sentence	Entity#1	Entity#2	Relation
Dhoni is the captain of Chennai Super Kings.	MSD	CSK	CaptainOf
Virat Kohli leads the Indian mens' cricket team	VK	IND	CaptainOf
Virat Kohli plays for Royal Challenger's Bangalore.	VK	RCB	PlaysFor
MS Dhoni is India's wicket keeper	MSD	IND	WKeeperOf
Dhoni keeps wickets for Chennai.	MSD	CSK	WKeeperOf
Kohli might leave RCB after the 2020 season	VK	RCB	<None>

Given an ontology and a sentence corpus, a Human Expert labels each sentence with the entities present in it and the relation between them(as per the sentence).

Note that the last example is provided for illustrative purpose, and if the expressed relation is not a part of the ontology, the Human Expert is likely to simple delete it.

DISADVANTAGES OF HUMAN SUPERVISION

- High quality human labelled data is expensive to produce and hence limited in quantity
- Because the relations are labeled on a particular corpus, the resulting classifiers tend to be biased toward that text domain
- Bootstrapping is possible, but due to limited and biased seeds, semantic drift is likely to take place

INTRODUCING DISTANT SUPERVISION

Distant supervision for relation extraction without labeled data

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky
Stanford University / Stanford, CA 94305
{mikemintz,sbills,rion,jurafsky}@cs.stanford.edu

DEFINING DISTANT SUPERVISION

For some ontology R, given

- A database D containing list of relations $\mathrm{r}(e_1, e_2)$, where $r \in R$, and $e_1, e_2 \in E$
- A corpus of natural language sentences S containing information about entities in E,

Output list of tuples $[\mathrm{r}(e_1, e_2), \mathrm{s}]$, where $\mathrm{r}(e_1, e_2) \in \mathrm{D}$, $s \in S$, and s expresses the relation r between e_1 and e_1

METHOD

1. Use a Named Entity Recognition tool to identify the entities participating in each sentence. If the entity count in any sentence is not equal to 2, or the discovered entities have no relation mentioned in the database, the sentence is discarded
2. For every sentence, if the named entities in it appear in some entry in D, add it to the training set for the corresponding relation.
3. Train a multi-class logistic classifier, which takes as input the features corresponding to a sentence, and outputs the relation between its two entities.

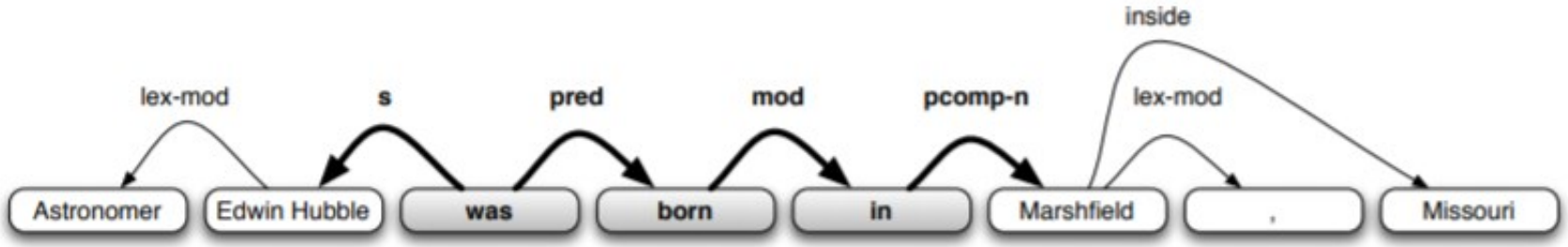
FEATURES FOR CLASSIFICATION

- Lexical Features(for k=0,1,2):
  - The sequence of words between the two entities
  - The part-of-speech tags of these words
  - A flag indicating which entity came first in the sentence
  - A window of k words to the left of Entity 1 and their part-of-speech tags
  - A window of k words to the right of Entity 2 and their part-of-speech tags
- Syntactic Features(for each "window node", ie , node not part of the dependency path):
  - A dependency path between the two entities
  - For each entity, one 'window' node that is not part of the dependency path
- The Named entity tag for both named entities.

FEATURES FOR CLASSIFICATION

Feature type	Left window	NE1	Middle	NE2	Right window
Lexical	[]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[]
Lexical	[Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[,]
Lexical	[#PAD#, Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[, Missouri]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]

Table 3: Features for 'Astronomer Edwin Hubble was born in Marshfield, Missouri'.




HUMAN SUPERVISION
I CAN EXPLAIN
MINTZ ET AL.
HOW COME YOU WORK IN LINEAR TIME WHILE I NEED POLYNOMIAL TIME?

PROBLEMS WITH THIS FORMULATION

- Multiple relations could exist between the same two entities. Like in our example, Dhoni is the captain as well as wicket-keeper for Chennai. These two relations are independent in general, but this model would put both sentences as training examples for both relations.
- Any corpus is likely to have sentences which do not contain any information(atleast as far as the ontology is concerned) about the relation between the entities it mentions.

PROBABILISTIC GRAPHICAL MODELS

Probabilistic graphical models (PGMs) are a rich framework for encoding probability distributions over complex domains: joint (multivariate) distributions over large numbers of random variables that interact with each other.

PGM's represent random variables as nodes in a graph, with edges representing dependencies between these variables. Depending on whether the edges are directed or undirected, two types of PGM's are most useful:

- Markov Networks(Undirected)
- Bayesian networks(Directed)

FACTORS

A factor is a function $\phi(X_1, X_2, \dots, X_k) = r \in \mathbb{R}$ where each X_i is a random variable.

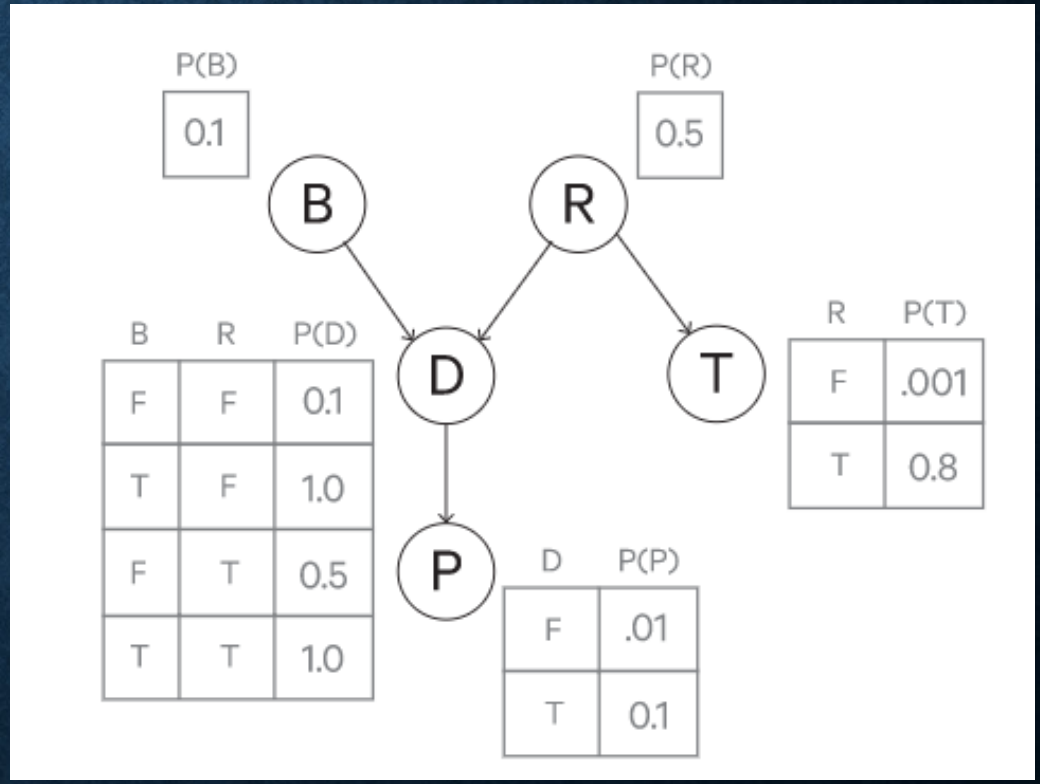
The set of random variables $\{X_1, X_2, \dots, X_k\}$ is known as the scope of the factor.

There are two primary operations defined on factors:

- A **factor product** of two factors ϕ_1 having scope $S_1 = \{Y_1, Y_2, \dots, Y_k, X_1, X_2, \dots X_l\}$ and ϕ_2 having scope $S_2 = \{Z_1, Z_2, \dots, Z_m, X_1, X_2, \dots X_l\}$ has scope $S_1 \cup S_2$ and is defined as
  $$\phi_1 \times \phi_2(y_1, \dots y_k, z_1, \dots, z_m, x_1, \dots, x_l) = \phi_1(y_1, \dots y_k, x_1, \dots, x_l) \times \phi_2(z_1, \dots z_m, x_1, \dots, x_l)$$
- A **factor marginalisation** is similar to a probability marginalisation, but applied to factors

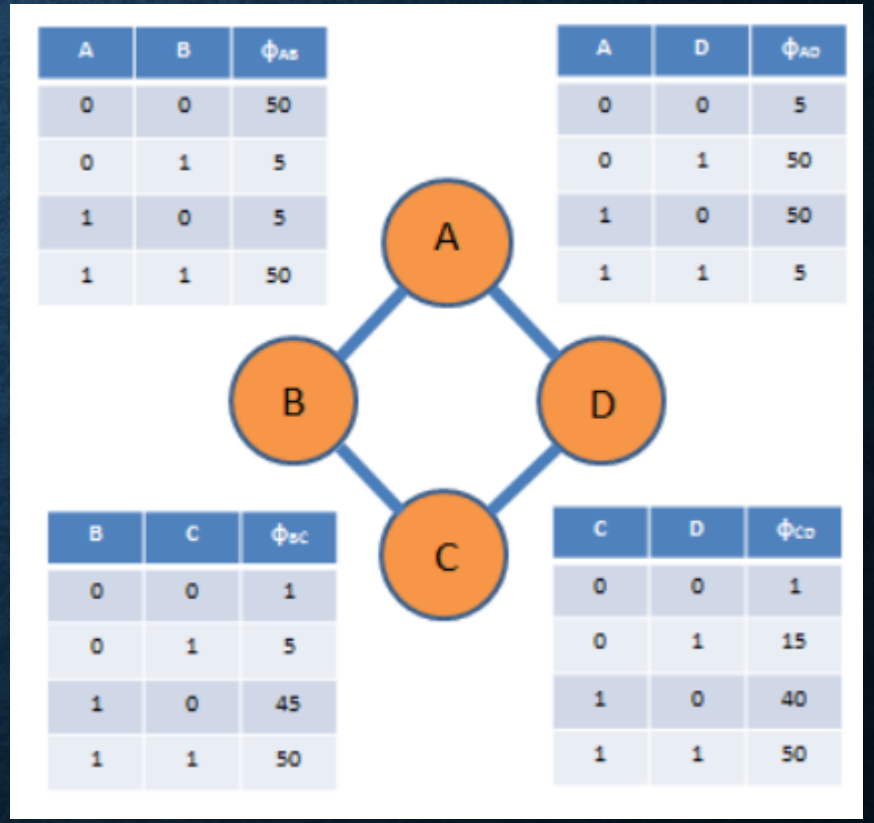
BAYESIAN NETWORKS

- In a Bayesian network, all edges are directed, and an edge from X_1 to X_2 indicates that X_2's probability distribution depends on the value taken by X_1
- Since dependencies cannot be circular, A Bayesian network graph must be acyclic
- Each node has a factor that lists the conditional probabilities of each state of that node, given the states of each of its parents.



MARKOV NETWORKS

- In a Bayesian network, all edges are undirected. An edge between two nodes indicates that the states of their respective variables affect each other.
- Each edge has a factor having scope equal to the nodes it connects. It lists the relative stability of every possible configuration of the variables. Sometimes, we might also have factors over cliques instead of edges.
- The factors themselves have no real interpretation in terms of probability. Multiplying all factors together and normalising gives the joint distribution over all variables



PGM'S AND INDEPENDENCE

- Amongst the many interpretation's of PGM's, one is to say that PGM's represent free as well as conditional dependencies and independences between a set of random variables.
- Two variables are independent(dependent) if information cannot(can) flow between their respective nodes.
- To check conditional independence/dependence, complete information is assumed at all nodes which are being conditioned upon

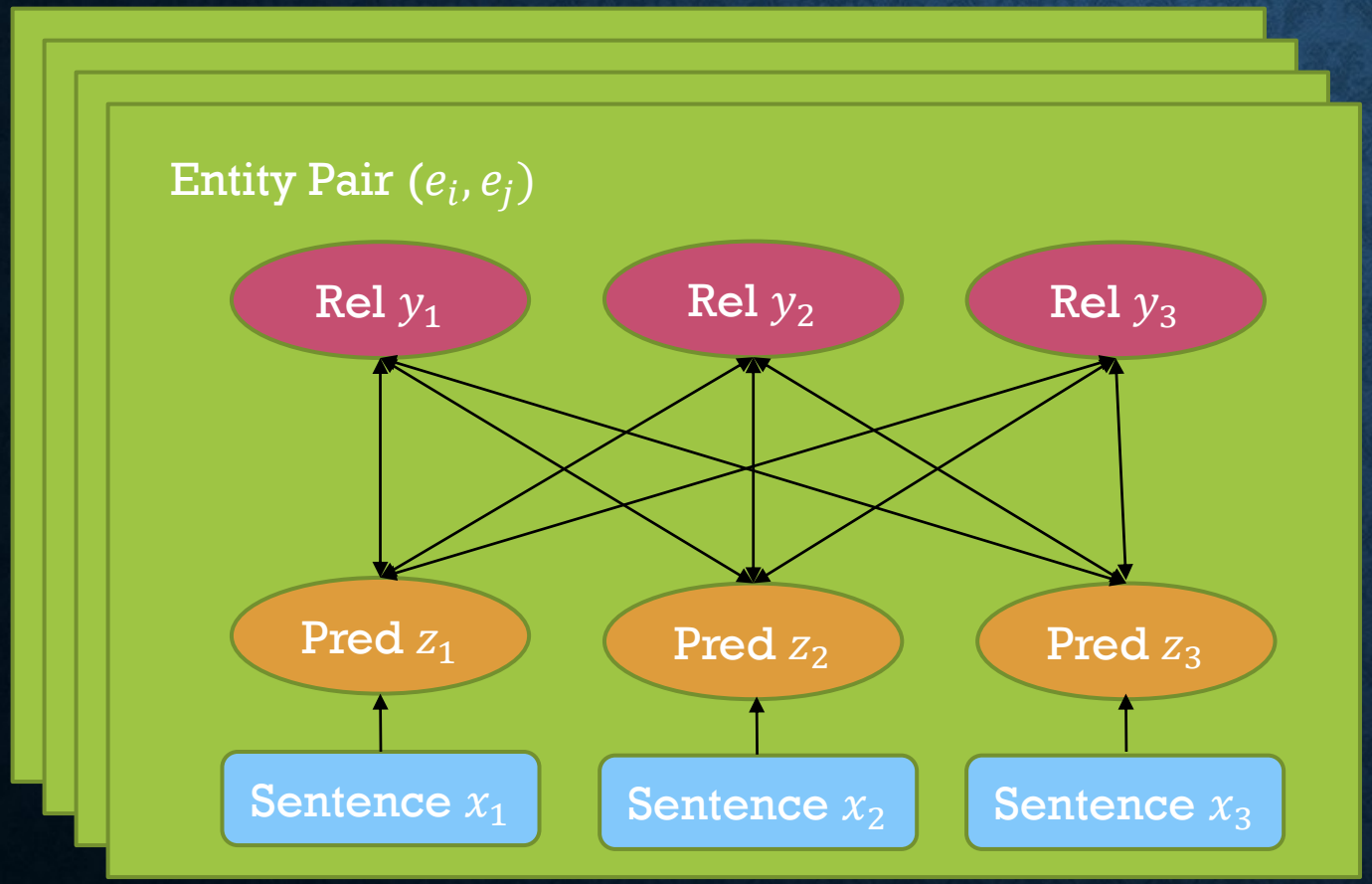
INFORMATION FLOW

- In a Markov network, information flowing in to a node through an edge can flow out through any edge unless we have complete information on that node
- In a Bayesian Network, information flow is slightly more involved:
  - Information flowing in through an outgoing edge can flow out through any other edge unless there is complete information on that node
  - Information flowing in through an incoming edge can flow out through an outgoing edge unless there is complete information on that node
  - Information flowing in through an incoming edge can flow out through an incoming edge only if there is some information on that node.

CONVERTING BETWEEN MARKOV NETWORKS AND BAYESIAN NETWORKS

- Two probabilistic graphical models are equivalent if they represent the same set of free and conditional independences
- With the exception of some special cases, it is impossible to find a markov network that is equivalent to a given Bayesian Network
- It is however possible to convert a given Bayes Net to a Markov Net that conveys independences that are a subset of the independences conveyed by the Bayes Net, such that the set of excluded independences are as few as possible. This is done by a process known as moralisation
- Converting a Markov net to Bayes net is much harder.

A PROBABILISTIC GRAPHICAL MODEL OF THE SCENARIO



- There is a different plate for each entity pair that appears in some relation in the database D. All factors are shared across plates. On each plate, there is a y node corresponding to each relation type in the given ontology. These nodes are binary, and take value 1 iff the given entities satisfy the current relation.
- There is an x node for each sentence in the corpus. It lies in the appropriate plate. It's value is the set of features discussed earlier.
- There is a z node corresponding to each x node. It's value ranges over all relation types in the given ontology, and it takes the value corresponding to the relation expressed it its sentence. xz factors are

REVISITING MINTZ' DS

In light of the Graphical model on the previous slide, we can think of Mintz' as follows:

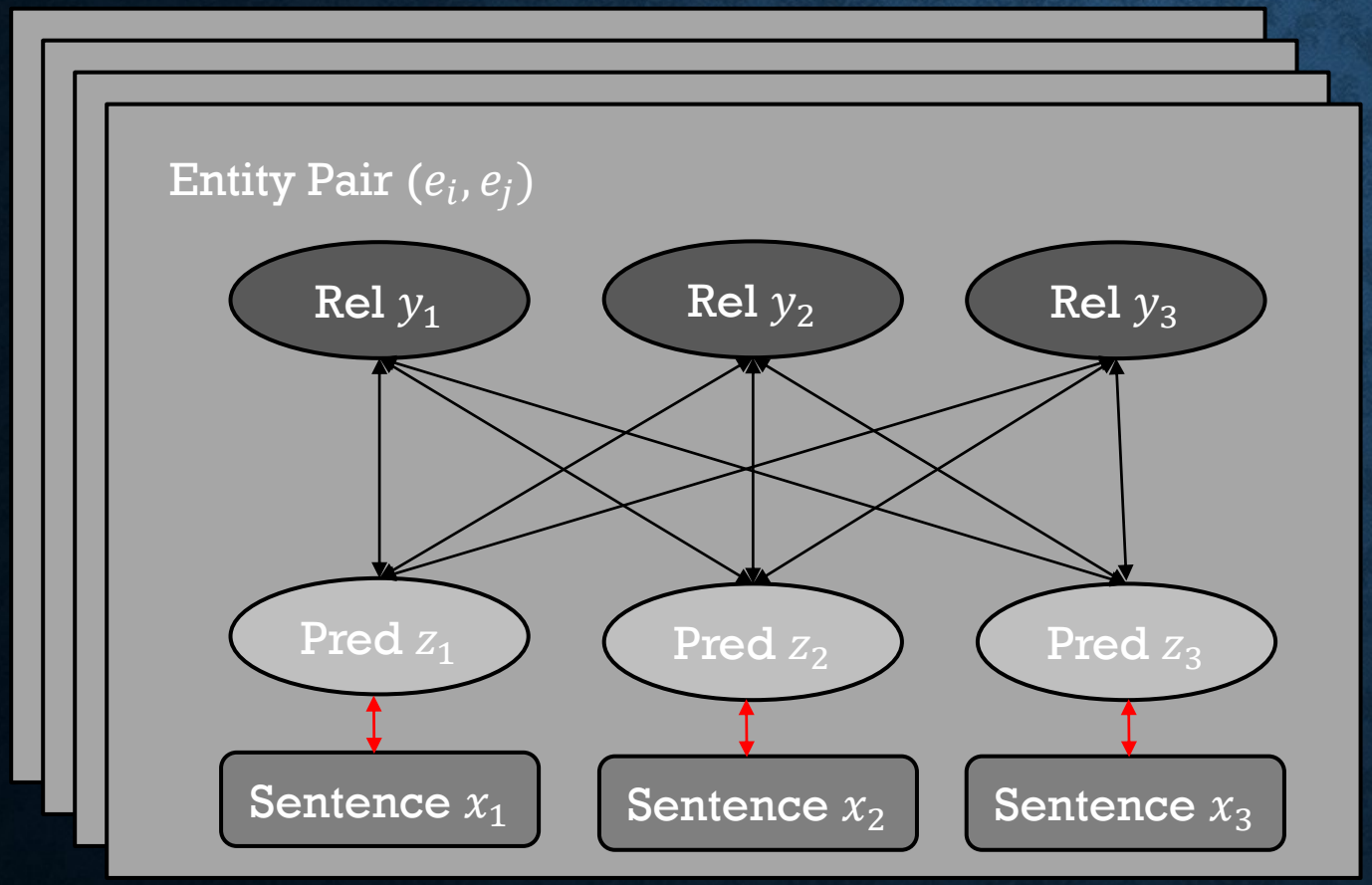
- All sentences across all plates share common factors for the (x, z) relations.
- Assuming that only one y is true in each plate, all z's on that plate must have value equivalent to the index of that y
- If more than one y is true on a plate, the model breaks down.

ALLOWING OVERLAPPING RELATIONS

Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, Daniel S. Weld
Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
{raphaelh,clzhang,xiaoling,lsz,weld}@cs.washington.edu

METHOD

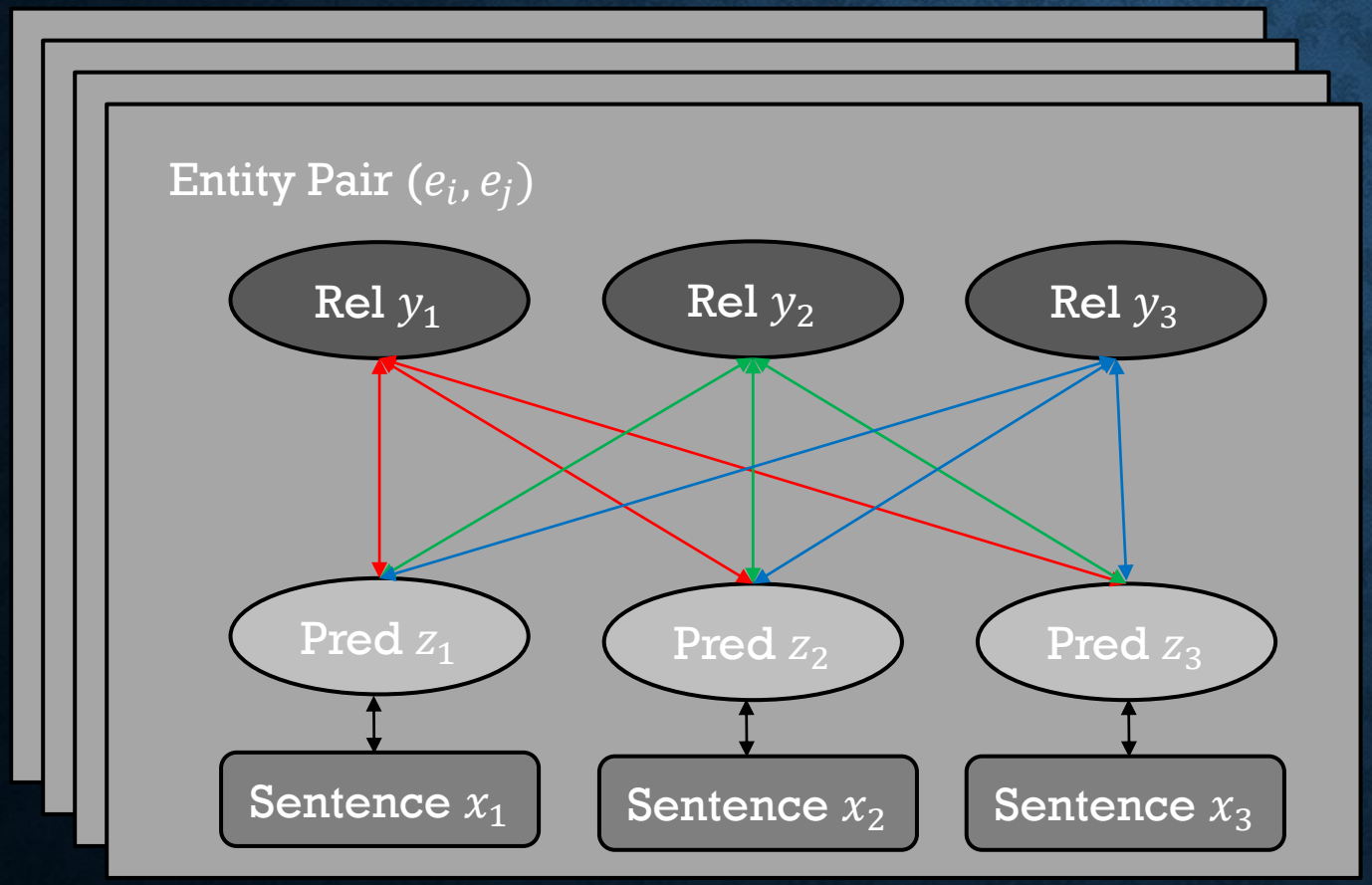


The xz edges(marked in red) are made undirected. This makes the graph a Markov network

As before, the factors over these edges are approximated by multiclass logistic regression

The z nodes are now allowed to also take the value $< none >$ if the corresponding relation does not exist in the database

METHOD



All edges coming into a given y node share the same factor. This factor has value 1 if any of the z nodes takes the value corresponding to the y node in the factor. This is also known as the *Deterministic Or factor.*

In the adjoining figure, all edges of the same colour(red, blue or green) would share a factor.

METHOD

The joint distribution over z and y is expressed as:

$$P(\boldsymbol{Z} = \boldsymbol{z}, \boldsymbol{Y} = \boldsymbol{y} \,|\boldsymbol{X}; \theta) = \frac{1}{\mathcal{Z}_{\boldsymbol{X}}} \prod_{r=1}^{|R|} \Phi^{join}(y_r, \boldsymbol{z}) \prod_{i=1}^{|S|} \Phi^{extract}(\boldsymbol{x_i}, z_i)$$

Where, $\Phi^{join}(y_r, \boldsymbol{z}) = \begin{cases} 1 & if\ y_r \wedge \exists i \text{ st } z_i = r \\ 0 & otherwise \end{cases}$

And $\Phi^{extract}(\boldsymbol{x_i}, z_i) = e^{\sum_j \theta_j \phi_j(z_i, \boldsymbol{x_i})}$, where ϕ_j are the features

The objective is to maximize the likelihood of $\boldsymbol{y}$ given $\boldsymbol{X}$


HUMAN SUPERVISION
I CAN EXPLAIN
MINTZ ET AL.
HOW COME YOU WORK IN LINEAR TIME WHILE I NEED POLYNOMIAL TIME?
HOFFMAN ET AL
I AM NP COMPLETE
imgflip.com

APPROXIMATIONS

- Instead of optimizing the whole objective at once, the algorithm runs in an online fashion, considering one plate at a time. The logarithm of the pointwise objective has the following derivative:

$$\frac{d\log(O_i(\theta))}{d\theta_j} = E_{\{\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x_i}, y_i; \theta)\}}[\phi_j(\boldsymbol{x_i}, \boldsymbol{z})] - E_{\{\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x_i}; \theta)\}}[\phi_j(\boldsymbol{x_i}, \boldsymbol{z})]$$

- Further, using Viterbi approximation, the expectations in the above equation are replaced by maxes.

ALGORITHM

initialize parameter vector $\Theta \leftarrow 0$
for $t = 1...T$ **do**
  for $i = 1...n$ **do**
    $(\mathbf{y}', \mathbf{z}') \leftarrow \arg\max_{\mathbf{y},\mathbf{z}} p(\mathbf{y}, \mathbf{z}|\mathbf{x_i}; \theta)$
    if $\mathbf{y}' \neq \mathbf{y_i}$ **then**
      $\mathbf{z}^* \leftarrow \arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x_i}, \mathbf{y_i}; \theta)$
      $\Theta \leftarrow \Theta + \phi(\mathbf{x_i}, \mathbf{z}^*) - \phi(\mathbf{x_i}, \mathbf{z}')$
    end if
  end for
end for
Return Θ

Calculating the first argmax is easy, because yz dependencies are all deterministic. This is equivalent to saying which *z* is most likely given the sentences on the plate.

The second argmax is somewhat harder, and can be reduced to a weighted edge cover problem, for which polynomial time algorithm is known.

Here, n is the no of plates and T is the no of iterations

PROBLEMS WITH THE FORMULATION

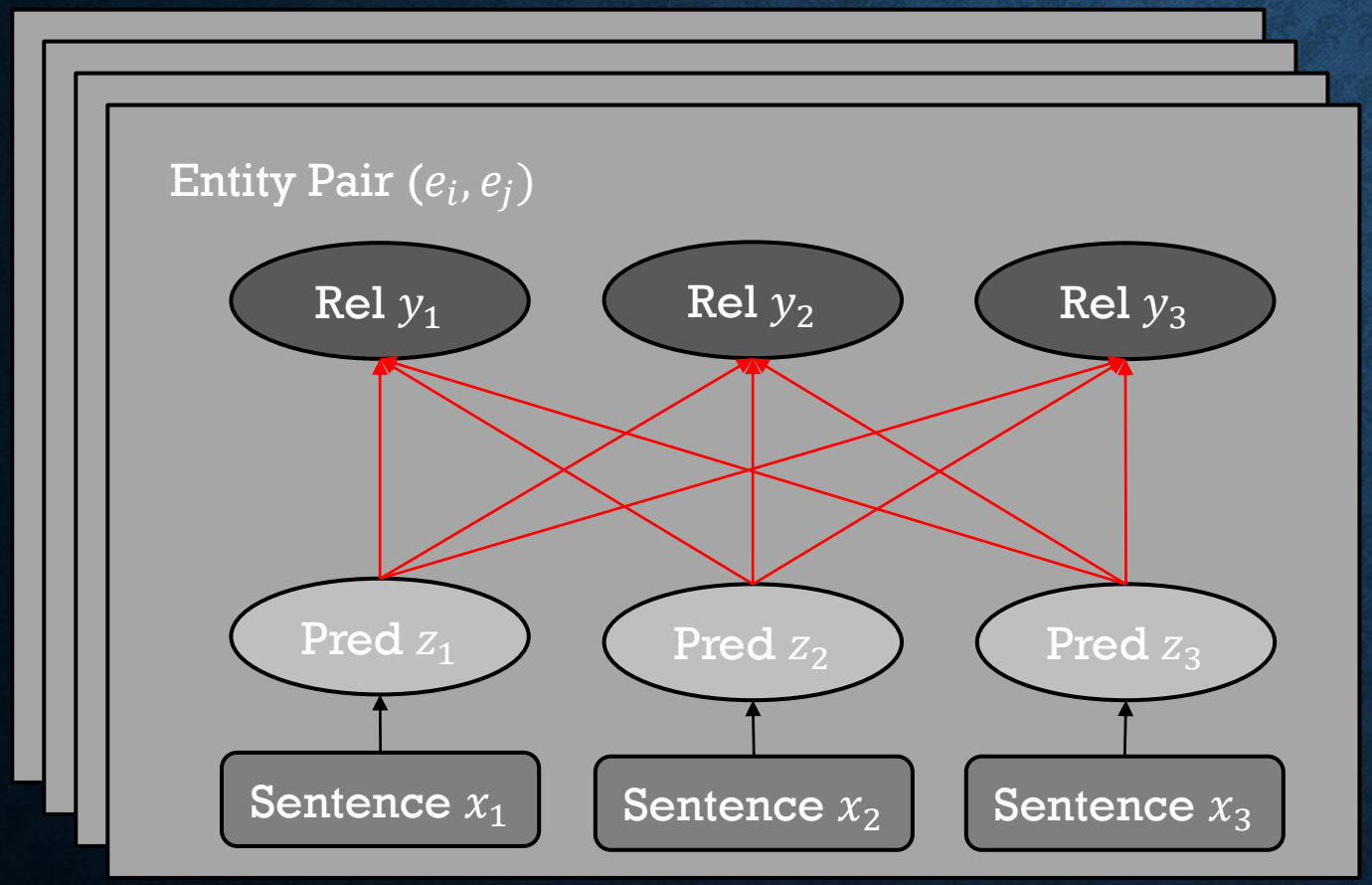
- All y nodes are frozen with no flexibility. This leaves no scope for the model to extract facts which are true and mentioned in the corpus, but do not occur in the database
- Frozen y nodes do not allow the model to do any inference over relation types, like if two relation types tend to be generated simultaneously.
- Deterministic yz factors disallow situations where a certain fact is mentioned in the database but does not occur in the corpus.

BAYESIAN DISTANT SUPERVISION

Multi-instance Multi-label Learning for Relation Extraction

Mihai Surdeanu[†], Julie Tibshirani[†], Ramesh Nallapati[*], Christopher D. Manning[†]
[†] Stanford University, Stanford, CA 94305
{mihais,jtibs,manning}@stanford.edu
[*] Artificial Intelligence Center, SRI International
nallapat@ai.sri.com

METHOD

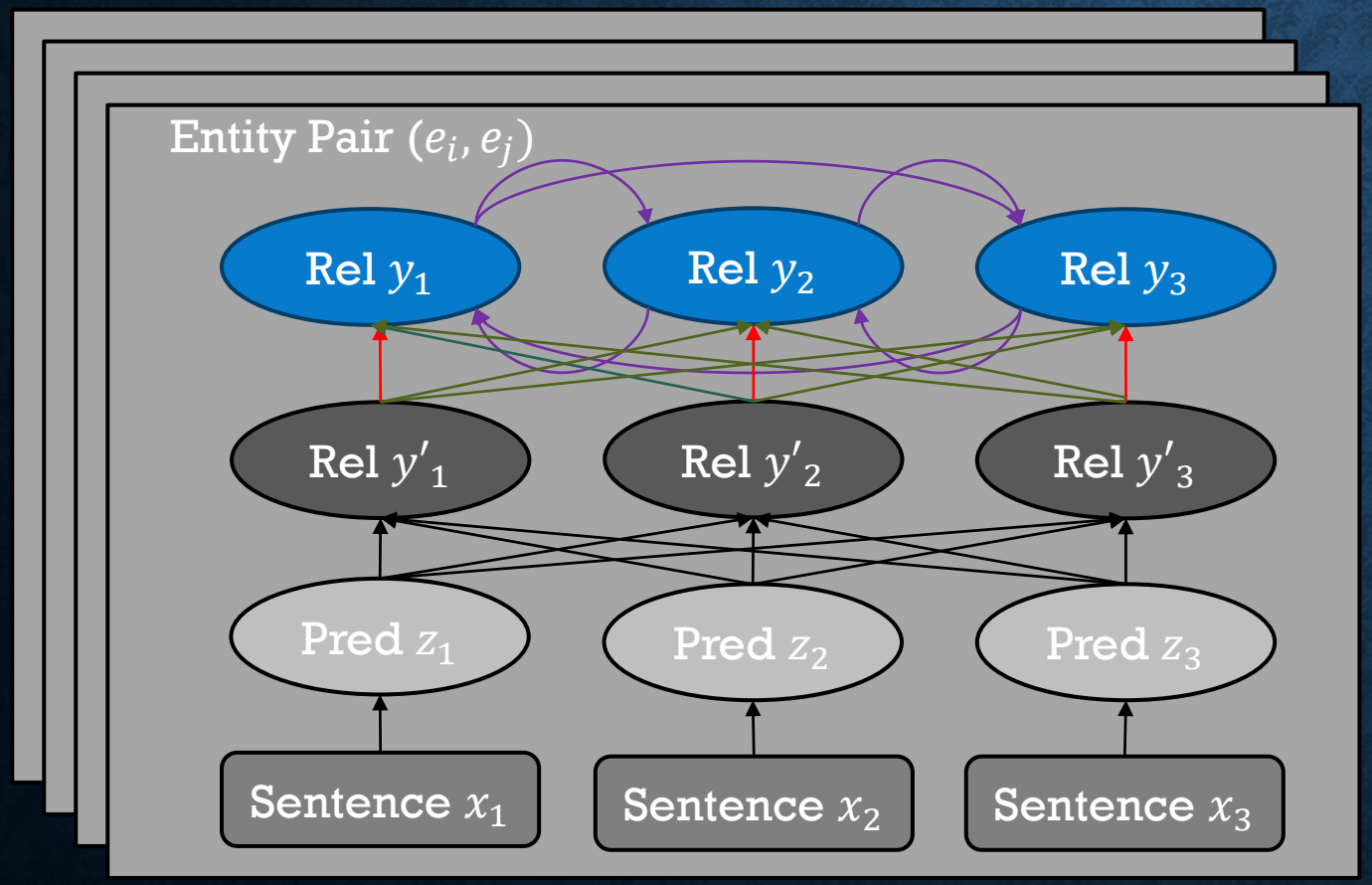


All yz are made directed from z to y. This leaves us with a Bayesian network.

Technically speaking, this modification means that if the value of any y node was known(even partially), then all the z nodes of that plate become correlated.

The xz dependencies remain unchanged, and are modelled by multi-class logistic regression as before.

METHOD



A new layer of nodes(blue) is added. From now on, this layer will be referred to as y and the original y layer will be referred to as y'.

The connections zy' connections still have deterministic or factors. Each y' node is connected to the corresponding y node.

The new y nodes share factors across all plates. These are learnt using binary logistic classifiers. The parameters for these classifiers will be referred to as $\boldsymbol{W}$.

OBJECTIVE FUNCTION

In accordance with Bayesian Networks, the joint probability over z and y for the i^{th} plate can be factorised as:

$$P(\boldsymbol{Z_i} = \boldsymbol{z_i}, \boldsymbol{Y_i} = \boldsymbol{y_i} \,|\boldsymbol{x_i}; \boldsymbol{\theta}, \boldsymbol{W}) = \prod_{\boldsymbol{m}} P(\mathrm{z_{im}}|\mathrm{x_{im}}; \boldsymbol{\theta}) \prod_{\boldsymbol{r}} P(y_{ir}|\mathbf{z_i}; \mathbf{W})$$

The aim is to optimise the log likelihood of $\boldsymbol{\theta}$ and $\mathbf{W}$ for the known values of $\boldsymbol{X}$ and $\boldsymbol{Y}$. This can be expressed as:

$$\boldsymbol{\mathcal{LL}}(\boldsymbol{\theta}, \boldsymbol{W}) = \sum_{\boldsymbol{i}} log(P(\boldsymbol{y_i}|\boldsymbol{x_i}; \boldsymbol{\theta}, \boldsymbol{W})) = \sum_{\boldsymbol{i}} log(\sum_{\boldsymbol{z_i}} P(\boldsymbol{y_i}, \boldsymbol{z_i}|\boldsymbol{x_i}; \boldsymbol{\theta}, \boldsymbol{W}))$$

Where the last term is the joint probability mentioned above. This objective is maximised using the EM algorithm


HUMAN SUPERVISION
I CAN EXPLAIN
MINTZ ET AL.
HOW COME YOU WORK IN LINEAR TIME WHILE I NEED POLYNOMIAL TIME?
HOFFMAN ET AL
I AM NP COMPLETE
SURDEANU. ET AL
YOU GUYS CAN ACTUALLY SOLVE IT??
imgflip.com

EXPECTATION STEP

In the expectation step, we select the most likely values to all the latent variables. In our case, we want to do this for z. Ideally, this would be as follows:

$$\boldsymbol{z}_i^* = \underset{\boldsymbol{z}_i}{\operatorname{argmax}}\, P(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}, \boldsymbol{W})$$

However, since this is intractable, we break this objective over each sentence. Further, since $\boldsymbol{y}_i$ is fixed in this step, we can write:

$$\begin{aligned} P(z_{im}|\boldsymbol{y}_i, \boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{W}) &\propto P(z_{im}, \boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{W}) \\ &\approx P(z_{im}|x_{im}; \boldsymbol{\theta}) P(\boldsymbol{y}_i|\boldsymbol{z}_i'; \boldsymbol{W}) \\ &= P(z_{im}|x_{im}; \boldsymbol{\theta}) \prod_{\boldsymbol{r}} P(y_{ir}|\boldsymbol{z}_i'; \boldsymbol{W}) \end{aligned}$$

Where $\boldsymbol{z}_i'$ is the previous value of $\boldsymbol{z}_i^*$ with the m^{th} index replaced with z_{im}

MAXIMIZATION STEP

In this step we optimize the parameters to better suit the current state of variables. In out case, the parameters under question are $\boldsymbol{W}$ and $\boldsymbol{\theta}$. These are optimised independently:

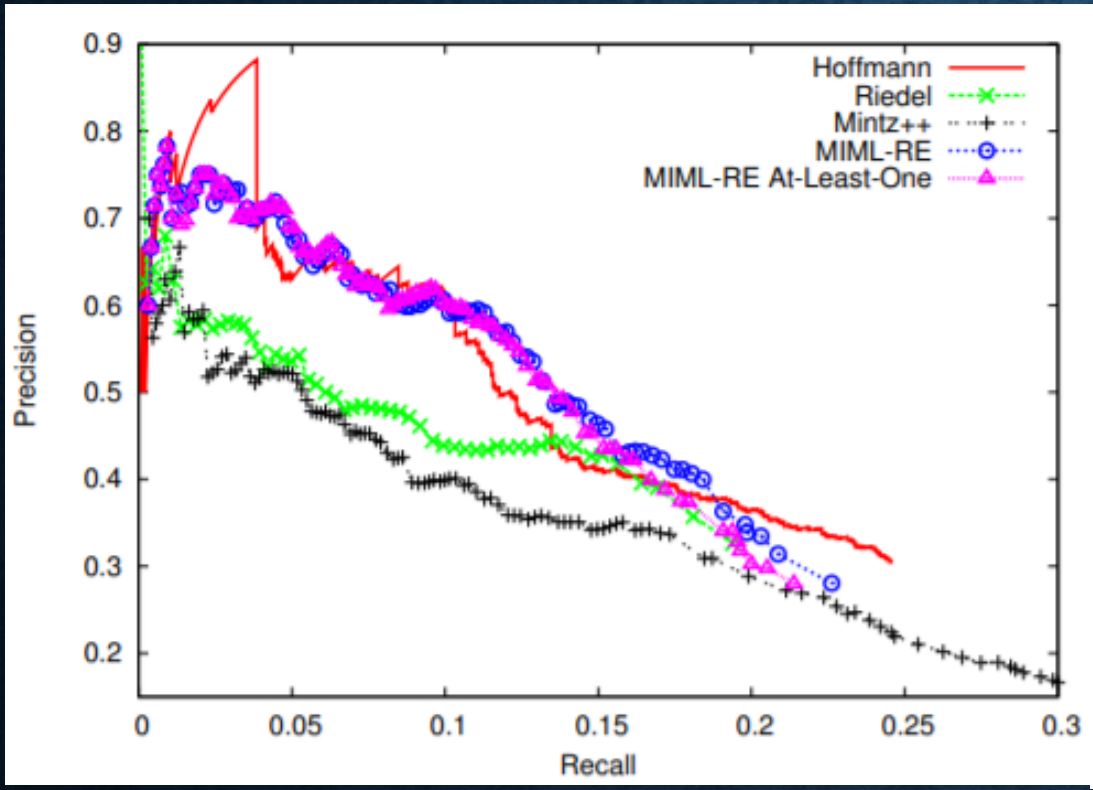
$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\boldsymbol{i}} \sum_{\boldsymbol{m}} P(z_{im}^{*} | x_{im}; \boldsymbol{\theta})$$

$$\boldsymbol{W_r} = \underset{W_r}{\operatorname{argmax}} \sum_{\boldsymbol{i}} \sum_{\boldsymbol{r}} P(y_{ir} | \boldsymbol{z}_{\boldsymbol{i}}^{*}; \boldsymbol{W_r})$$

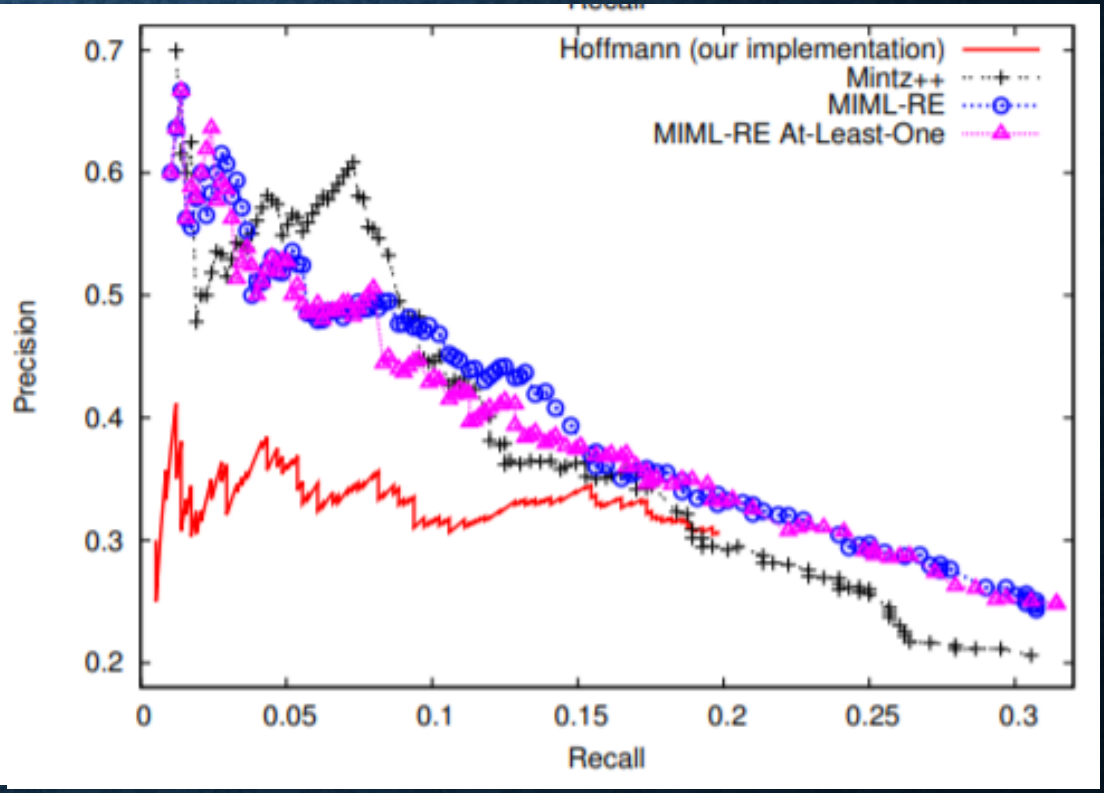
IMPLEMENTATION SPECIFIC DETAILS

- **Initialisation:** Since the model involves optimising a large set of parameters over a non-convex objective, a good initialisation is very important. For this purpose,
  - θ is initialised using Mintz et al's algorithm
  - W is initialised using the Hoffmann et al's algorithm.
- **K-fold training:** To avoid overfitting on the data, the data is split into multiple folds. For each fold, the classifier that runs the E step on that data runs the M step on the rest of the data only. The final classifier is generating by averaging all classifiers.
- **Randomisation:** In E-step, each sentence uses the modifies z value for all sentences encountered before it. Since this can lead to bias, the order or sentences is randomised across iterations

RESULTS



Riedel Dataset



KBP Dataset

MAJOR POSITIVE COMMENTS

- Great Mathematical Detail(Atishya, Pratyush, Siddhant, Saransh, Lovish etc.)
- Very clear about algorithm, hyperparameters, initialisation strategies. Easily replicable(Rajas, Siddhant, Saransh, Lovish etc)
- Modeling interactions(Rajas, Atishya, Soumya, Jigyasa), and therefore correct mistakes(Keshav, Shubham)
  - It is unclear to me how the relation level classifier is able to correct any mistakes, since the y variables never get updated during the training procedure
  - A running point was about whether constraints should be hard or soft. However, since the model is probabilistic and there are no deterministic factors involved, all constraints are soft.

MAJOR NEGATIVE POINTS 1

- Features/techniques handcrafted/not general(Rajas, Atishya, Soumya, Pratyush, Siddhant, Jigyasa, Lovish)
  - Features are mostly picked up as is from other papers. Those looking for justification might want to look into those papers. If not detailed there as well, I believe it is a fault of those papers, not this one
  - The main selling point of this paper is their model according to me. Therefore, it makes sense to use previously known features for various datasets. Using different features might also make comparisons unclear
- Strongly dependent on initialisation(Keshav, Pratyush, Pawan), Convergence of EM(Shubham)
  - The authors do a good job of reducing the problem to an EM framework. Queries such as these have been well addressed in EM literature.
- Noise/incompleteness of KB may affect Solution(Keshav, Rajas, Jigyasa, Shubham, Lovish)
  - Since the entire model is probabilistic, it already has scope for noise/incompleteness
- Does not handle multiple relations in sentence(Keshav, Rajas, Soumya, Shubham)
  - Possible future work direction. Might make sense to preprocess sentences using CALM for some cases.(Soumya)

MAJOR NEGATIVE POINTS 2

- Asymmetric dependence between relations(Pratyush)
  - As far as I understand, this asymmetry has been modelled. All dependencies are directed, so there are separate parameters for $y_1 \Rightarrow y_2$ and $y_2 \Rightarrow y_1$
- Improvement not much(Pratyush, Siddhant)
- Not scalable to Extreme classification(Siddhant), Too many parameters(Soumya), Imbalanced classes(Jigyasa, Saransh)
  - It is unlikely for ontologies to get very large, since they are hand-crafted. Too many parameters should never be an issue, as all the three papers agree on one thigh: there is an abundance of sentence corpora as long as you are somewhat flexible on their quality. Since there is an abundance, even skewed classes are likely to have abundance of examples
- Exponentially many choices for latent variable(Pawan)
  - There will always be a tradeoff between expressibility and computability. The paper does a good job of handling this in my opinion

EXTENSIONS 1

- Knowledge Base Completion(Keshav, Rajas, Atishya)
  - As said before, I feel this can be achieved if y nodes are also optimised in the E step
- Neural Classifiers(Atishya, Jigyasa, Shubham)
- Confidence in classification(Soumya)
  - Ideally should be handled by the confidence of classifiers in the current model
- Multiple relations in sentence(Keshav, Rajas, Soumya, Shubham)
  - Hierarchal Learning(Pratyush)
  - Using attention(Pratyush)

EXTENSIONS 2

- Inference over Knoowledge base for completion(Pratyush)
- Extending Ontologies(Siddhant)
  - I feel this is more of a downstream task
- Use Knowledge graph embeddings(Jigyasa, Saransh)
- Contradicting sentences(Pawan)
  - Subjective knowledge is not really knowledge, and should not be made a part of knowledge base. However, if there seems to be agreement on one side, then the current model should be able to handle it probabilistically
- Additional layer for top-k embeddings(Lovish)
  - ~~Seriously? You want to increase the degrees of freedom for an already intractable model?~~ Might make sense, but this layer must be deterministic

QUESTIONS?

Thank you