A Social Network Platform Architecture Based on Markov Logic and Transformer Based Neuron Translation

Bokai Xu
School of Data Science
The Chinese University of Hong Kong, Shenzhen
bokaixu@link.cuhk.edu.cn

Zhuoyu Li
School of Data Science
The Chinese University of Hong Kong, Shenzhen
zhuoyuli@link.cuhk.edu.cn

Abstract

In this work we seek to realize a new kind of social network platform which could automatically make large scale and concrete inference of underlying truth of social issues on social network platform. To accomplish this, we introduced the Jianfeng Social Network: a new kind of social network platform architecture based on Markov Logic and Transformer based neuron translation. We carefully discussed the technical details of such an architecture. To realize the core functions of such a platform, we utilize Markov Logic Network, a probability distribution model combined with first order logic (FOL). We also utilized the technology from Natural Language Processing, including Named Entity Recognition, Transformer based neuron network translation models. Our architecture is partially based on Deep Learning, and well combined with causal inference. We also created new concepts, such as Cumulative Predicates Library, which could help remove bias of inference result. We demonstrated that such an architecture could successfully handle real world issues by using a toy example. The performance is similar to the reasoning of human with logical thinking training. Finally, we developed a social network platform to realize our architecture. However, the learning process of Markov Logic Network is P-complete. And there is no mature Transformer based model from natural language to first order logic. So, the technology is not feasible for large scale problems yet. We leave it as future work to realize approximate learning and inference algorithm in Markov Logic Network. We also leave Transformer based neuron translation model to future work, and replace it with artificially designed algorithm.

Contents

1 Introduction

1.1 Background: Why We Want to Build Such a New Kind Of Social Network

The outbreak of New Coronavirus Pandemic in late 2019 and early 2020 attracted some attention on social networks when the outbreak was in its nascent stage, but due to algorithmic flaws in today's social networks, we were not able to make timely judgments and speculations about the pandemic. A review of social media messages from the early stages of the outbreak shows that some people were already aware of the details and others were already alerting people to the outbreak at least until December 15. By December 27, the classification of the virus had already been determined. The local hospital in Wuhan could not have been unaware of the facts either, but for various reasons, this information was not used effectively.

Learning from the lessons of the epidemic, we wanted to develop a new kind of social network that could make full use of the information provided by users, weigh the relationships between the various pieces of information, and make use of existing a priori knowledge, while being able to learn and evolve itself. Therefore, we chose Markov Logic Networks to do this.

Combined with the natural language processing technology, we were able to reach a wide range of users and ensure that the platform is easy to understand.

We call our new social network Jianfeng in respect to our statistical inference instructor.

In this demo, we have implemented most of the features of the above ideal social network using the Python flask framework. All the natural language processing, inference process, is dynamic rather than static.

In this demo, we use a relatively small problem scenario.

The Albert, Bob, David gang steals a final exam paper and causes a huge backlash on the school's social network. We want to use what people say on the social network to infer who stole the test and who lied about it.

1.2 The Properties of Markov Network and the Origin of Our Idea

Markov Network is an undirected graphical model for representing dependencies between random variables.

A Markov network can be represented by an undirected graph $G = (V, E)$ where the nodes in V represent random variables and edges in E represent dependency relationships.

Let us consider X, a kind of assignment of values to the variables in a Markov Network. We call C the set of maximal cliques in the network and assign a factor ψ_c to each clique $c \in C$. The probability $P(X)$ is,

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{(k)}) \tag{1}$$

We know that Markov Network can be used to represent a system, where this system is in general form, because the random variables in V can be either numerical or categorical.

If we want to represent a world in the form of Markov Network, we can consider binary case, for example, if Tom lied, we can assign **True** to the random variable **lie(Tom)**. So each random variable in our desired network can only take one of the two binary values **{True, False}**.

The world above is an assignment of values of V.

We are continuously thinking about the way to infer the truth of the event. PageRank algorithm by Jimmy Page gave us the confidence to utilize graph theory to solve the truth of an event. We tried PageRank Algorithm to infer the truth of an event by assigning the relationship **{Entailment, Contradiction, Independence}** between any pairs of comments on the social network. Then we construct a directed graph of this set of comments. Each comment will post its importance to the comment that it entails, and we can construct a transition matrix, and then calculate A^{500} or above, until the matrix converges. Then we can find the stationary point. Then we can rank the

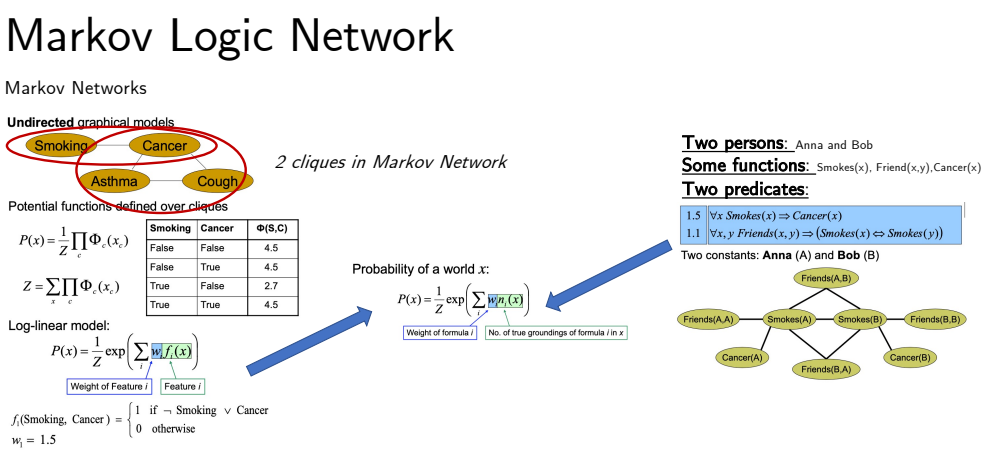
comments according to their PageRank value. In this process, we observe the specific property of graph. However, this algorithm may not work very well in practice, because its time complexity is $O(n!)$. Now we focus on the combination of graph and First Order Logic.

We find that un-directional graph may perform better than directional graph. Because the computer cannot really understand what you mean at a extremely high precision, and the sentences generated by users may not entail each other by our definition. So, to use First Order Logic and build an un-directional graph may be better than PageRank algorithm when we analyse the comments by netizens.

Markov Network is a joint probability distribution of many variables, the distribution can be decomposed into several cliques according to conditional independence. Each clique has a potential function to model its local probability. The joint probability is the product of all potential functions.

It can be further written as a log linear model, then we map the coefficient w_i to weights and potential function f_i to whether a predicates is satisfied.

We use the mapping rule to construct a Markov Network (such a markov network is a Markov Logic Network)and do inference on it, then we can get the inference result.



Markov Logic Network can be viewed as a interpretation of Markov Network

Figure 1: How Markov Network is interpreted as a Markov Logic Network

1.3 Why We Use The First Order Logic

Atom clauses can be easily understood by programming languages, and it is similar to functions and thus can be easily processed by algorithms. In this project, we aim to convert every natural language sentence into atom clauses of First Order Logic. For example, **Tom accuses Bob of stealing** can be converted into First Order Logic expression . And this form can be utilized by Markov Network as we mentioned above, because can be assigned a value between **{True, False}**. The combination of First Order Logic and Markov Network can utilize the strength of two models.

However, we have to mention that the First Order Logic can be used with Markov Network in two ways: The first role is the representation of events. The second role is to help us determine the possible Markov Network. Why? Because the assignment of values of Markov Network is not known. We have to determine which possible way of assignment is more probable. Then we consider some

constrains. These **contrains** are also written in First Order Logic form and can help to contruct Markov **Logic** Network.

1.4 The Nature of Markov Logic Networks

We can view Markov Logic Network (MLN) as the template of Markov Network. MLN is underdetermined and uncertain. We need to determine the variable value. However, because the value is difficult to determine, MLN uses constrains in First Order Logic (FOL) to select those most feasible world. Worlds with more conflicts with stronger constrains has a low probabilty of existence. So MLN meets our demand very well.

1.5 The Components of Markov Logic Networks

Before we get started, we shall first define the components of MLN. MLN consists of four parts represented in FOL:

1. **Knowledge Base** Some facts about this possible world, written in FOL. For example, "Tom accuses Bill of stealing the money" is represented as "accuseofstealingmoney$Tom, Bill$".
2. **Function Declaration** Some functions mentioned in the world, written in FOL. For example, "accuseofstealing$person, person$". This function has two inputs, the first argument is a person and the second argument is also a person.
3. **Named Entities** Some entities that belong to some specific categories. For example, "person = {Tom, Jerry, Bob, Nazarbayev}".
4. **Predicates** $Constrains$ Some general rules without pointing out any named entities, objects in the constrians are represented as x,y,z, and some underdetermined objects.
   For example, steal_the_money(x) $\wedge$!accuse_of_stealing(x,y) then lie(y) means "if x stole the money and y did not accuse x of stealing money y lied."

2 Knowledge used in the Project

2.1 Natural Language Processing

Natural language processing (NLP) is a field concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them [**?**].

We have to deal with the natural language records so that we have to choose some NLP algorithms. The most important part is to do text segmentations, transfer the texts into first order logic and atomic sentences to match the requirements of the Markov Logic Network.

2.2 Markov Logic Networks

The markov networks model used in this project comes from the article written by Richardson and Domingos in 2006 [**?**]. The below statements in this subsection are from the article.

A Markov network (also known as Markov random field) is a model for the joint distribution of a set of variables $X = (X_1, X_2, \cdots, X_n) \in X$. The joint distribution represented by a Markov network is given by

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{(k)}) \tag{2}$$

where $x_{(k)}$ is the state of the kth clique (i.e., the state of the variables that appear in that clique). Z, known as the *partition function*, is given by $Z = \sum_{x \in X} \prod_k \phi_k(x_{(k)})$.

2.3 Markov logic network

A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = c_1, c_2, \cdots, c_{|C|}$, it defines a Markov network $M_{L,C}$ (Equations 1 and 2) as follows:

1. *$M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is* 1 *if the ground atom is true, and* 0 *otherwise.*
2. *$M_{L,C}$ contains one feature for each possible grounding ofeach formula F_i in L. The value ofthis feature is* 1 *if the ground formula is true, and* 0 *otherwise. The weight ofthe feature is the wi associated with F_i in L.*

All the formulas and the constants in the Markov Logic Networks have to meet the 3 assumptions below:

1. **Unique names.** *Different constants refer to different objects.*
2. **Domain closure.** *The only objects in the domain are those representable using the constant and function symbols in (L, C)*
3. **Known functions.** *For each function appearing in L, the value of the function applied to every possible tuple of arguments is known, and is an element of C.*

3 Packages used in the Project

3.1 Natural Language Processing

The NLP package used in this project is **AllenNLP** [**?**]. We use the methods of AllenNLP to do text segmentations, transfer the texts into first order logic and atomic sentences so that we can match the requirements from the Markov Logic Network.

3.2 Markov Logic Networks

Pracmln is a toolbox for statistical relational learning and reasoning and as such also includes tools for standard graphical models [**?**]. We use this package to build the markov logic networks and give the results.

3.3 Flask

Flask is a micro web framework written in Python [**?**]. We use **Flask** to show the demo of social network comments section. From the **Figure 2**:

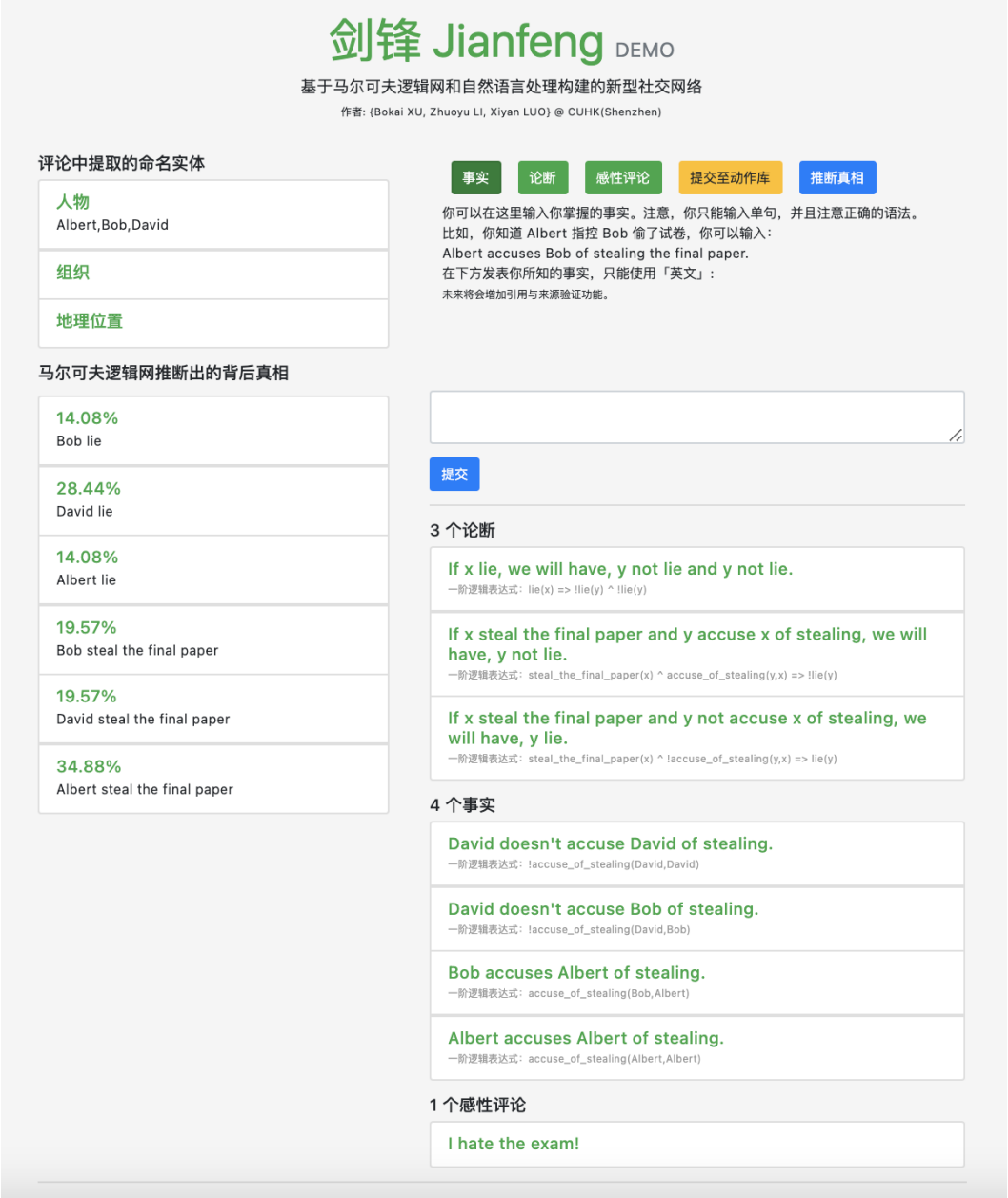


Figure 2: An screenshot of the realized social network system

we can find that we can input and post comments in a text area. We even can choose the type of the comments. However, if we choose the wrong type, the NLP model will indentify it and change the type into the correct one. All of the comments will be showed below the text area. Also, the names and functions in the comments will be showed in the left sides. The result of the Markov Logic Networks will be showed below them.

4 The Architecture of Jianfeng Network

4.1 Users Can Post 3 Kinds of Comments

Difference from traditional social network, here users can choose 3 different columns to post their comments: *Facts, Predicates, Emotional*.

For *Facts* column, users can post the facts they have mastered. For example, if the user know that *Albert accuses Bob of stealing the final paper*, he or she could type *Albert accuses Bob of stealing the final paper*. To make sure that the facts are valid, citation and source verification features will be added in the future.

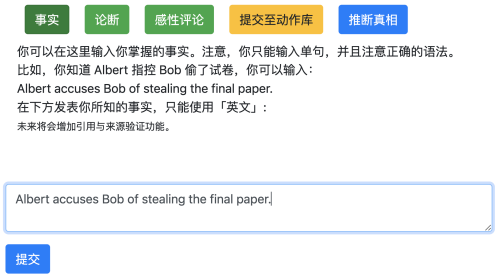


Figure 3: Prompting Users to Post facts

For *Predicates* column, users can post their own judgements and theories. These predicates are general and can reveal their thoughts. To make sure the meaning of the logic expression is correct, we only accept First Order Logic expressions. We expect well educated users to post on *Predicates* column. To make this process more smooth, we provide a toolbox for our users. For example, we provide widely used logic operators, function library, undetermined objects.

Figure 4: Prompting Users to Post Facts

After users post their *Predicates*, their logic expressions will be stored in our system and we design an algorithm to convert the logic expression into natural language. We display the natural language version of the *Predicates* on the front end, other users can give *likes* to each *Predicates*. We will determine the *weights* of the *Predicates*. The *weights* of each *Predicates* are important in Markov Logic Network, they are the *strength* of constraints.

For *Emotional* column. In this column, users are welcomed to post everything they want. This column is an area for users to post their emotional comments, and if users post emotional comments in *Facts* column, the comments will be transferred to this column because we also design an algorithm to recognize if the comments are really *Facts* or *Predicates*.

4.2 For Less Educated Users

Jianfeng is desgined to the serve the public, and its users are mainly less educated users. Less educated users accounts for the majority of our users. So, we carefully designed the architecture of Jianfeng.

Figure 5: Displaying the Comments on Front End

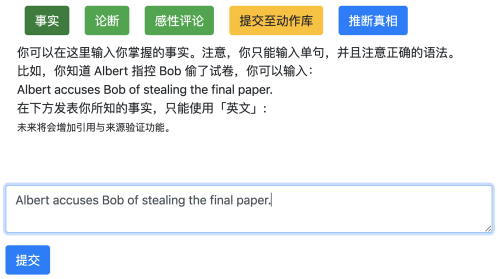


Figure 6: Prompting User to Commit Facts

Less educated users can post the facts they know just by typing English sentences.

The facts in natural language will be processed by AllenNLP Open Information Extraction module [cite: https://demo.allennlp.org/open-information-extraction] first. The result given by AllenNLP is as follows:

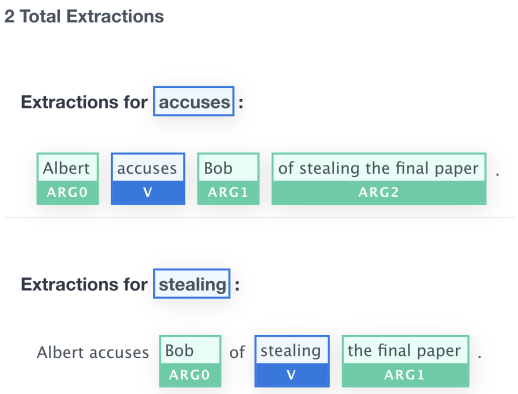


Figure 7: Primary Result Given by AllenNLP Open Information Extraction module

AllenNLP gives multiple results of one single sentence, and we want to find one that can best model that sentence. We choose the verb which can recognize most words as its arguments. We call that verb the best verb of that sentence.

Then we consider the best verb and its arguments. For example, we input the sentence *Albert accuses Bob of stealing the final paper.*, and the output has two possible results: *accuse* and *steal*. Then we choose *accuse* as our best verb because it can utilize 3 components as its arguments.

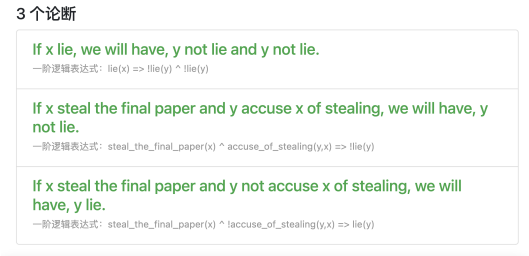


Figure 8: Natural Language Converted

The next step is to utilize AllenNLP Named Entity Recognition module to check each argument to see if it is a named entity, for example, *Albert* and *Bob* will be recognized as *Person*; *of stealing the final paper* will be not recognized. Then, we append *of stealing the final paper* to the verb and only keep *Albert* and *Bob* as arguments. Then we can construct the function $accuses\_of\_stealing\_the\_final\_paper$ with two input arguments *ARG0: Person* and *ARG1: Person*. Here we can express this fact as $accuse\_of\_stealing\_the\_final\_paper\,(Albert, Bob)$.

After we extract the function mode, we will first append this function to the library, in Jianfeng Demo, we call it *The Actions Extracted From User Comments*. But in consideration the experience of less educated users, we design an algorithm to convert functions into natural language expressions. For example, if the user submitted *Albert accuses Bob of stealing the money*, we will first extract the function mode and store it into *action library* and then compile it into natural language expression, then display it on the front end.

Less educated users can also post emotional comments. We give them a choice to post whatever they want. Users can choose *Emotional* module to post their comments.

We also design an algorithm to convert First Order Logic expression into natural language. Because some well educated users can post First Order Logic expressions and complex expressions, it is usually hard to read for less educated users. It is necessary to convert every piece of logic expression into natural language. And we realize this function in Jianfeng demo.

4.3 Cumulative Action Library

Jianfeng Demo will continuously collect actions submitted by users. Once user post a fact containing valid information, we will automatically analyze the action mode in this sentence. Then it will be stored in our *Action Library* and we will convert it into a form which can be easily understood by users and then display them on the front end.

Users can submit new *action modes* to *Action Library*. We provide a column for users to submit new mode. Users only need to provide a sentence in this dialogue, and they don't need think about the details. We will automatically analyze the sentence and extract the *action mode* in this sentence.

Besides, the *Action Library* in Jianfeng Demo is cumulative. We will keep all the *action modes* in our database, and once a new event emerges, users can conveniently use previously defined *action modes*.

We expect in a few months, we can collect all the possible action modes in human language. In that case, we will provide more complete toolbox for educated users who have a good command of First Order Logic to composite their predicates.

Furthermore, the *action modes* in consideration within an event comply with the following conditions:

1. That *action mode* is mentioned in *Facts*.

2. That *action mode* is mentioned in *Predicates*.

Action modes satisfy the above conditions will be declared in Markov Logic Network inference process (refer to Components of Markov Logic Network part).

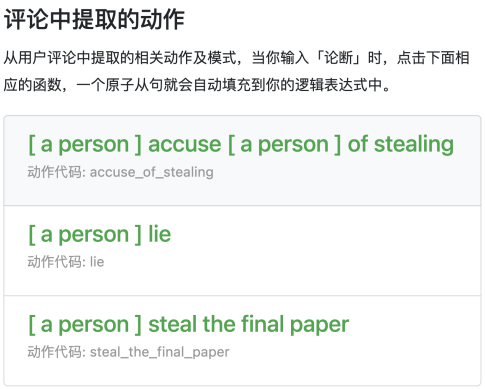


Figure 9: Natural Language Expression of Action Library

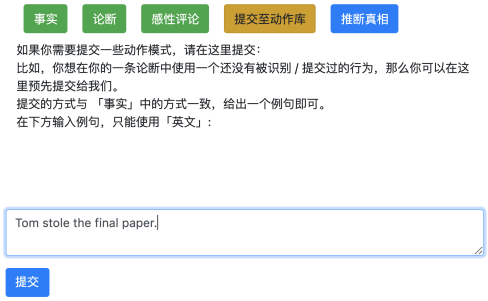


Figure 10: Action Mode Submission Dialog

4.4 Cumulative Predicate Library

The *Predicates* are of vital importance in Markov Logic Network, they can help to construct the whole network. Some general *Predicates* can be utilized in many cases automatically. That helps to improve the inference result.

Besides, because Markov Logic Network can learn from the final results, thus determining the *weights* of each *predicate*, it is important to determine the *weight* of each *predicate*. Jianfeng system is designed to perform MLN learning process once we know the posterior results. Then we can determine the *weights* of each *predicates* in the corresponding event. Because sometimes the learning result is not in $(0, 1)$. Then we use logistic function to convert them into $(0, 1)$. The next problem is to find the relative *weights* in case two irrelative *predicates* come together. To solve the problem, we should first operate Jianfeng system for months, collect enough scenarios. Then perform a global MLN training. Then we can get an overall predicate ranking list.

5 Example: Who stole the final exam paper

5.1 Background

The final exam paper was stolen. That event caused widely discussion on the campus BBS. Three students are suspected steal the final exam paper. And students have some information about that event.

5.2 Users' Speech on the Traditional Social Network

Albert post the following comments on social media:

I accuse myself of stealing the final exam paper!

Bob post the following comments on social media:

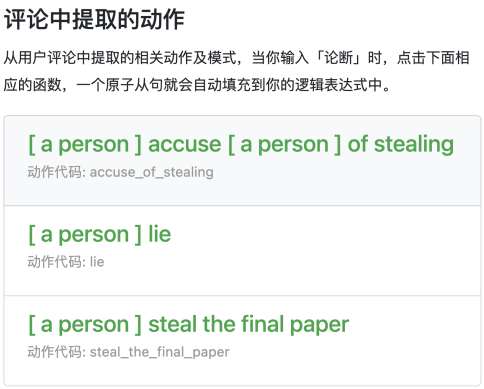


Figure 11: Natural Language Expression of Action Library

I accuse Albert of stealing the final exam paper!

David post the following comments on social media:

I don't accuse myself of stealing the final exam paper, and I don't accuse Bob of stealing the final exam paper.

On the usual social media platform like *lgulife.com*, users may post some very emotional comments. People will not come to a common view. They will feel confused after the discussion. Who stole the final exam paper?

5.3 What Can Jianfeng Platform Do: For Less Educated Users

Let's consider what less educated users can do on Jianfeng platform . Users can post the above information to the *Facts* column. Some students just post the information mentioned above to Jianfeng:

1. Albert accuses Albert of stealing.

2. Bob accuses Albert of stealing.

3. David doesn't accuse Bob of stealing.

4. David doesn't accuse David of stealing.

5.4 What Can Jianfeng Platform Do: For Educated Users

And some educated users who have good commands of First Order Logic can think about their theories. For example, Tom defined *what it means if a person lies* post to the *Predicates* column,

$$steal\_the\_final\_paper\,(x) \wedge !accuse\_of\_stealing\_the\_final\_paper\,(x,y) \Rightarrow lie\,(y)$$

That means, if a person A stole the final paper, and another person B did not accuse him or her of stealing final paper, then person B lied.

and another person Jim added to the definition of lying,

$$steal\_the\_final\_paper\,(x) \wedge accuse\_of\_stealing\_the\_final\_paper\,(x,y) \Rightarrow !lie\,(y)$$

That means, if a person A stole the final paper, and another person B did accuse him or her of stealing final paper, then person B did not lie.

That seems enough for the inference process.

5.5 What Can Jianfeng Platform Do: Inferring the Underlying Truth

Based on the above information, what can we infer?

Steve made a query about *Who lied, Who stole the final paper*. The inference result is,

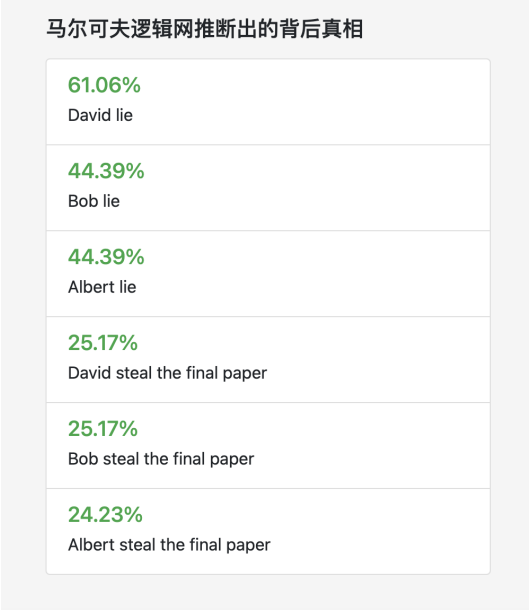


Figure 12: Underlying Result

We can find that Jianfeng platform thought of that:

1. David is the person that has the highest probability of lying.

2. Three persons are equally likely to steal the money.

So, it is obvious that the information provided is not yet enough. However, the basic common sense is formed.

5.6 User Interaction Presentation

The *Facts* column are:

Figure 13: Facts Column

The *Predicates and Logic* column are:

Also we can input some emotionals (Whichever the type we choose they will be shown as the emotionals finally):

Now, we can find that all of the names and functions appeared in the comments are clearly showed in the left side:

We click the button of running the Markov Logic Networks and find that the result is showed below in the form of probabilities:

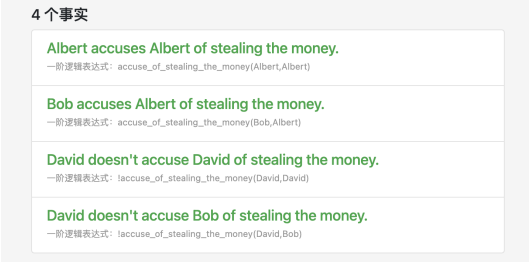


Figure 14: Predicates and Logic Column

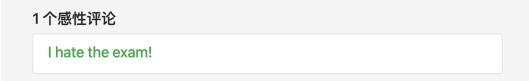


Figure 15: Emotional Column

6 Conclusion and further development

6.1 Conclusion

It is possible to build a social network that automatically reasons about things and presents user comments in a reasonable way. Our example shows that the Jianfeng platform can effectively integrate and leverage information from a variety of sources and fully exploit a prior knowledge and historical paradigms when dealing with the unknown event. For example, for public users, we adapt well to their needs. For users who have good commands of First Order Logic, we provide more opportunities for them to play. At the same time we make the information they post understandable to the public users.

As Rousseau said in *The Social Contract* , *general will* must exist, the only question is how to find them. Rousseau believed that full discussion must lead to the general will. We expect that the Jianfeng platform will be able to fulfill this wish in the future. At least, in this demo, we demonstrate the potential of the Jianfeng platform with a small, daily life example.

Figure 16: Extracted Named Entities and Action Mode

6.2 Further development

Back to the background of our project, the purpose of our project is to do some analysis of the outbreak of New Coronavirus Pandemic. However, duo to official strategy, the MLN does not give a obvious result on the conclusion. Also, the first order logic is not easy to be converted from the common social network comment sections and there are some comments are too complex so that the NLP does not works well too.

The further development on this project that we want to do is to use more strong tools of NLP to deal with the complex logic in the comments and find another way to compute the relationships between the complex logic. So far, the development of the NLP maybe unable to handle such powerful work like this and we may need more support on the more powerful models than MLN too.

A Appendix

A.1 Architecture of Jianfeng Network: Prototype

Based on the property of Markov Logic Networks (its input is facts, predicates, weight) and output is the probability of unobserved variables) We propsed the architecture of our social network platform. First, users can post anything in natural language on a specific topic. Then we use Transformer based neuro translation model to translate natural language into first order expressions (some posts are translated into predicates, some are translated into facts) And users can vote all the predicates, then we get the weight of different predicates.

Then we plug facts, predicates, and corresponding weights to MLN and get final inference result.

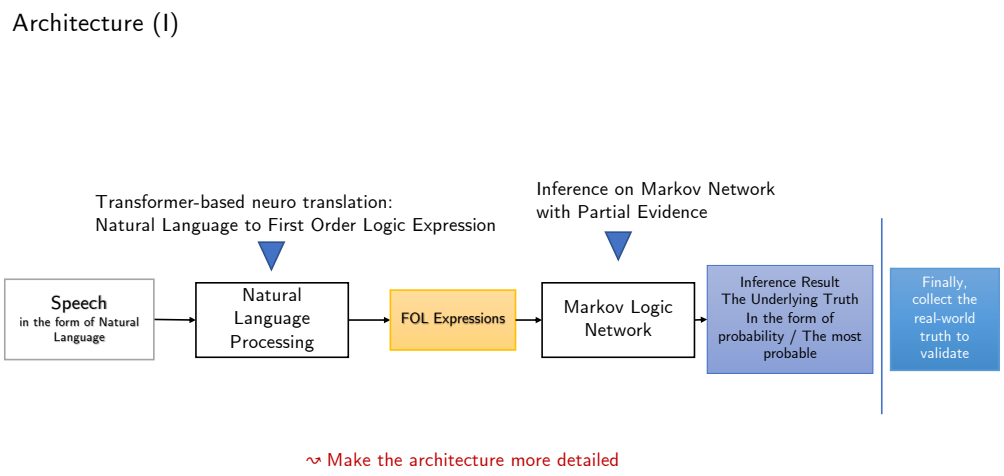


Figure 17: Prototype of Architecture

A.2 Architecture of Jianfeng Network: Detailed

Now we present a more detailed diagram showing the architecure.

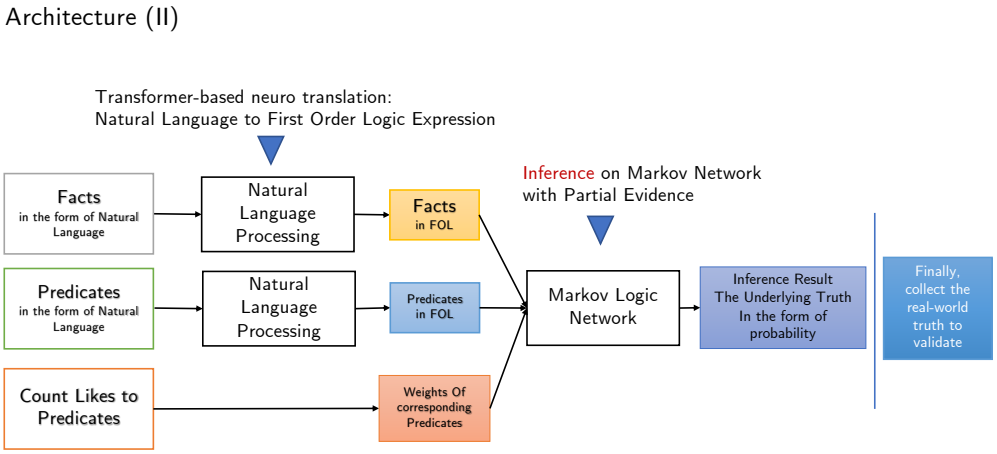


Figure 18: Detailed Architecture

A.3 Architecture of Jianfeng Network: Improvement for People's Bias

We have to mention that, sometimes the opinion of people is biased, so we have to improve the architecture. We can make use of the real-world truth. We can perform learning in such markov logic Networks to get a set of weights (also called parameters) and store them in a cumulative predicates database and reuse them in the future. The machnism can help the social network to make m,ore reasonable inference.

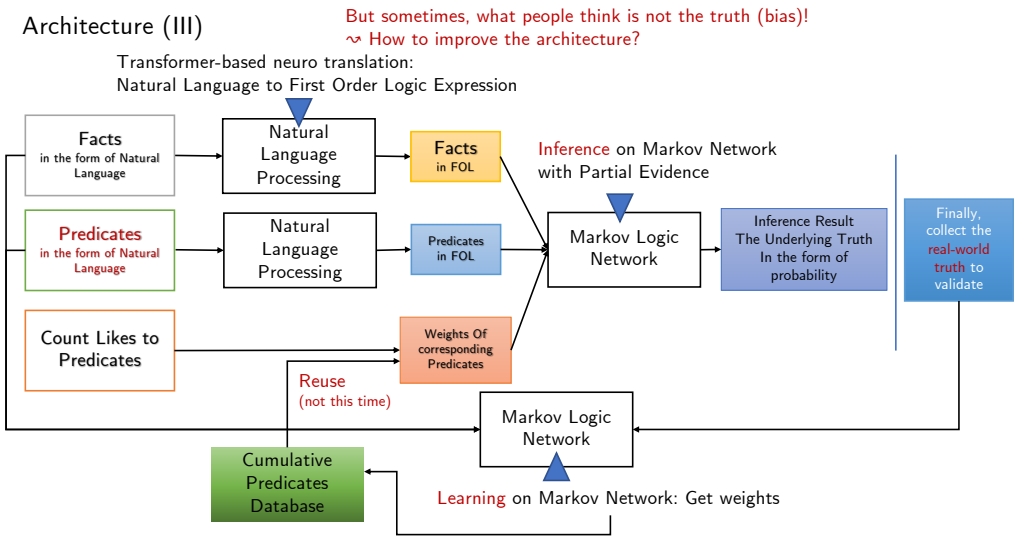


Figure 19: Improved Architecture

A.4 Architecture of Jianfeng Network: The Actual Architecture Realized in This Project

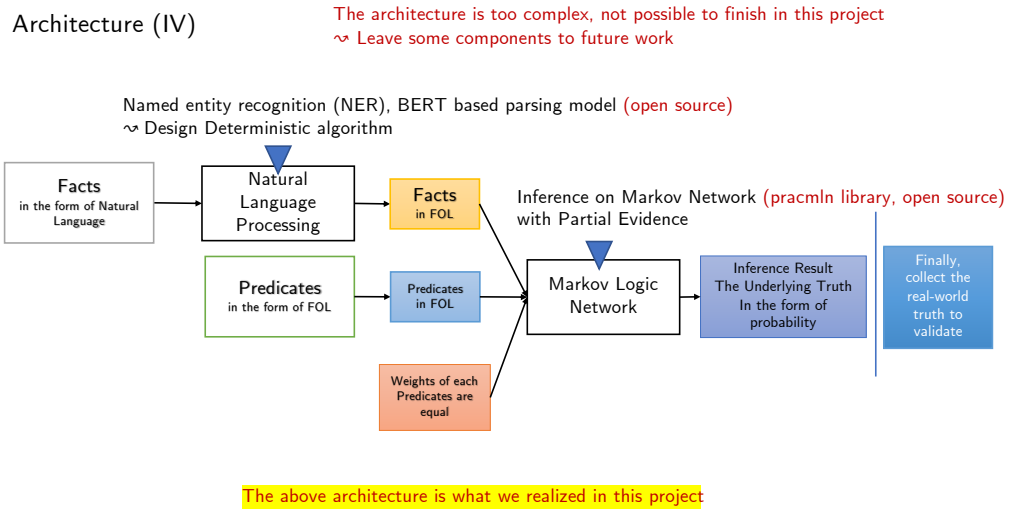


Figure 20: The Actual Architecture Realized in This Project