UNIVERSITY OF CALIFORNIA, SANTA CRUZ

MASTER'S PROJECT REPORT

User Expertise Detection in Online Communities

Author:
Dhawal JOHARAPURKAR

Chair:
Dr. Suresh LODHA
Reader:
Dr. Faisal NAWAB

A report submitted in fulfillment of the requirements
for the degree of Master of Science

in

Computer Science and Engineering

Fall, 2018

Declaration of Authorship

I, Dhawal JOHARAPURKAR, declare that this thesis titled, "User Expertise Detection in Online Communities" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

While detection of experts in online communities is a widely studied problem, most work fail to incorporate the inherent relational nature of these online communities. In this work, we explore the usage of the linkages that exist between posts and between users (structure) in these community question answering sites in the form of related posts, duplicate posts, etc. and use this relational structure in conjunction with local information to collectively predict users' expertise. We compare the statistics of the evaluation tests to the baseline results from related works to measure the performance of our approach compared to the non-relational methods. We see that using the relational structure of the data helps outperform baseline methods.

Acknowledgements

First and foremost, I'd like to thank Prof. Lise Getoor for her help and guidance in this project's ideation. I'd also like to thank Prof. Suresh Lodha and Prof. Faisal Nawab for their support and advice, Dr. Golnoosh Farnadi for all time we spent discussing and brainstorming and the members of LINQs lab for their inputs and help!

In addition, I'd also like to thank my family and friends for their constant support and presence in my life and through this process. And lastly, I'd like to thank the city of Santa Cruz and all it's people, this place has been a sanctuary!

Contents

Declaration of Authorship iii

Abstract v

Acknowledgements vii

1 Online Communities 1
1.1 Online Communities: A Contemporary Phenomenon 1
1.2 Understanding QA Communities 1
1.3 User Roles in Online Communities 2

2 Expert Identification in Online Communities 3
2.1 Why Is Expert Identification Important? 3
2.2 Variants of Expert Identification 3
2.3 Related Work 3

3 Need For Labelled Data 5
3.1 Dearth of Labelled Data in Online Communities 5
3.2 Data Programming 5
3.3 Related Work 6
3.3.1 Data Programming Related Work 6
3.3.2 Snorkel Related Work 6
3.3.3 Babble Labble Related Work 7
3.4 A New Paradigm For Data Labelling 7
3.4.1 Motivation 7
3.4.2 Data Programming 7
3.4.3 Snorkel 8
3.4.4 Babble Labble 9
Explanations 10
Semantic Parser 10
Label Aggregator 11
3.5 Takeaways 11

4 Relational Approach To Expert Identification 13
4.1 Probabilistic Soft Logic 13
4.1.1 PSL for Collective Classification 13
4.1.2 PSL for Expert Identification 14
4.2 Dataset 14
4.3 Model for Expert Identification 16
4.4 Results 17
4.5 Future Work 17

Bibliography 19

List of Figures

3.1 An overview of the Snorkel system 9
3.2 An overview of the Babble Labble system 10
3.3 Rule based semantic parser of the Babble Labble system 10

4.1 Distribution of comments vs threads 15

List of Tables

4.1 Dataset Stats 14
4.2 Dataset Statistics: considering users >10 posts (answers/question) . . . 16

Chapter 1

Online Communities

1.1 Online Communities: A Contemporary Phenomenon

In this chapter, we discuss the history of online communities and understand some of their dynamics, in an attempt to shed light on the objectives behind this work.

Online communities originally appeared in the 70s, when there were online games containing virtual worlds and were multi-player, and consisited several elements of role-playing games such as hack and slash, player versus player, interactive fiction, and online chat (Bartle, 1990). Players could carry out several functions common to today's online communities such as enter and leave rooms, read descriptions of rooms, objects, other players, non-player characters, etc. and interacted with other users through commands in a natural language like language.

The online commmunity movement was further fueled by the community The Whole Earth 'Lectronic Link (The Well) (Rheingold, 1994). The Well, allowed for its users to share their stories and knowledge in general subject areas known as conferences, which reflected member interest and allowed them to collaborate by creating conversational threads. The Well is a highly influential source for most online communities with modern day discussion forums and question answering communities are inspired from the concepts originally proposed in The Well.

Over the last two decades, the knowledge market has further surged with the emergence of more contemporary communities such as Yahoo! Answers, Quora, StackExchange and some of the now-discontinued communities such as Google Answers. (Hsieh and Counts, 2009) showed that a real-time market based Question Answering (QA) service where users would use their reputation as currency, by offering them to those who answer their questions, motivated users to participate more actively. More recently, microblogging services have popped up, and serve as a different type of online community as compared to the traditional QA styled ones.

1.2 Understanding QA Communities

Question answering communities allow their users to exchange and archive knowledge in the form of questions and answers, usually in a threaded fashion. A great amount of knowledge sharing occurs this way with popular questions gaining traction quickly and rising to the top and receiving plenty of answers in a short span of time (Adamic et al., 2008).

The interactions that occur on these threads is generally muti-faceted, ranging from expertise sharing, to discussions, advice, and support. Unlike online communities, QA communities don't have a bow-tie structure and lack a strongly connected cluster of users, and generally users either ask questions or answer questions, but not both. Regular users tend to ask more questions than answer them, and expert users tend to answer more questions than ask them.

(Shah and Pomerantz, 2010) studied the quality of answers in online QA communities and found that even when there is no monetary/reputation investment in answering a question, the answers generally far exceed the quality that would have been provided by library reference services.

1.3 User Roles in Online Communities

Different kinds of users assume different roles in online communities. Some contribute to knowledge, while others mostly consume knowledge. There are also users who try to vandalize communities and also those who act as administrators and moderators. As in the real world, their motivations may be complex: both the desire to be helpful and the desire to be noticed may prompt the writing of a lengthy eposition.

For most participants, embellishing their identity, in both ways - establishing their own reputation and gaining the recognition of others, plays a vital role. Thus, interactions of users, through actions performed on various posts (questions, answers, comments) such as upvoting, downvoting, marking as duplicate, referencing are vital signs of expertise, and end up forming a graph, inferring over which can lead to a highly sucessful process of detecting expertise.

Chapter 2

Expert Identification in Online Communities

2.1 Why Is Expert Identification Important?

As stated in previous sections, experts are important users in an online community, they take on several roles and ensure smooth functioning of the community. Some communities have several commemorative embellishments added to profiles of such users such as badges, reputation scores, extra services and/or abilities. In other words, most communities survive and thrive around these experts, as they make the core of a successful community.This work is centered around identifying such users and effectively identify them.

2.2 Variants of Expert Identification

Needless to say, identifying experts is a hard task, since there are several factors to this problem, and the dynamicity of which only renders it more difficult.

There are several expert-finding works out there that tackle this problem, observing them under different lenses:

- Early Identification of Potential Experts: This deals with identifying experts early on, so as to retain them and have the community benefit from their presence

- Human factors of expertise: This deals with studying the human psychology and the behaviours that drives certain people to become experts

- Dynamic changes in expertise: This deals with evolution of experts over time in a community, since the roles of an expert change over time and adapt to the needs and status of the community at different times

2.3 Related Work

In their paper, Pal and Konstan (Pal and Konstan, 2010) explore the identification of experts in online communities and also model user behaviour and user evolution. They introduce “selection bias", which is the bias exhibited by experts in selecting questions to answer, only selecting those to which they feel they can contribute to. They also say that an expert’s selection bias is consistently higher than a non-expert’s. Movshovitz-Attias, Steenkiste and Faloutsos (Movshovitz-Attias et al., 2013) state that majority of the questions are asked by non-experts, and the primary source of answers and the high-quality answers are experts. Attiaoui, Martin and Yaghlane (Attiaoui, Martin, and Yaghlane, 2017) introduce a metric called “Belief Measure of Expertise (BME), which is a more holistic measure than just reputation which then they use to predict experts.

The notable graph-based approaches are the PageRank algorithm (Brin and Page, 1998) and HITS algorithm (Kleinberg et al., 1999). Both of these approaches model the Internet as a graph in which a web page is represented by a node, and a directed link between nodeA and nodeB indicates that the web page corresponding to nodeA has a hyperlink that points to web page corresponding to nodeB. This representation of the Internet is called a web graph. (Brin and Page, 1998) proposed PageRank - a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page in which a web page with higher probability is considered more important than a web page with lower probability. This forms the basis of ranking web pages and is used successfully in the Google search engine. The HITS algorithm (Kleinberg et al., 1999) computes two values for a web page: its authority value, which estimates the value of content of the page, and its hub value, which estimates the value of its links to other pages. Authority and hub values are defined in terms of one another in a mutual recursion.

These graph based approaches are what we try to employ in an online community setting, since a lot of the features carry over in these settings.

Chapter 3

Need For Labelled Data

3.1 Dearth of Labelled Data in Online Communities

A lot of the cited works in this report have stated that they either used an existing pre-dated dataset(s) or have had to create a dataset from scratch, therefore raising a concern for the need of labelled data in these forums. Since a lot of predictive machine learning tasks require access to a labelled dataset, a lack of which, has created a demand for a process to create them.

3.2 Data Programming

This section is a survey on the recent progress in the improvement of data labelling processes. I study the newly introduced "Data Programming", a paradigm which deals with programmatic creation of datasets. Then, I look at "Snorkel", a system that allows users to train models without hand labelling any data. The users can write labelling functions representing heuristics to label data, and can vary in coverages and accuracies. "Snorkel" denoises their outputs and combines them to provide probabilistic data labels. Then, I study "Babble Labble", which is an extension built using Snorkel, that processes heuristics described in natural language to labelling functions and thereby used to label data. These works constitute an important foray in this space of data labelling, and is right on the horizon of the massive data wave currently under way.

Since most end-use machine learning models are discriminative, they suffer from the dearth of labelled data. This lack of availability of labelled data is due to several domain specific reasons, but also because it's difficult to collect, expensive, and might constantly change based on new discoveries or improvement in understanding of the space. The body of work that I study tries to sidestep this issue, by processing the unlabelled data and providing labels and this data generation process also lends itself into the model making phase, thus enabling the production of state-of-art models, quickly and cheaply.

In short, I survey three works:

- Data Programming: A new paradigm for the programmatic creation of training sets called Data Programming ((Ratner et al., 2016)).
- Snorkel: A system that enables users to train machine learning models without manually labelling data, by writing labelling functions representing heuristics ((Ratner et al., 2017)).

- Babble Labble: A framework for training classifiers using natural language explanations provided by annotators for each labelling decision ((Hancock et al., 2018)).

These works originate from Stanford's DAWN project, whose mission is to democratize AI by simplifying the building of AI applications.

3.3 Related Work

3.3.1 Data Programming Related Work

This work is a continuation of previous work in machine learning, termed as distant supervision. For example, in relation extraction from text, the input corpus is heuristically mapped to a knowledge base containing relations between entities (Mintz et al., 2009) (Craven and Kumlien, 1999). There are various extensions to these methods, such as multiple instance learning one, discriminative feature based models, generative models, etc. This approach is similar to the generative models, but differs in that the generative models are not built on the user's inputs, which is the case with this approach. There are other approaches (Shin et al., 2015) (Mallory et al., 2015) that use user's heuristics to directly label unlabelled data, but they don't deal with the noise generated by this labelling functions.

Other similar approaches include crowd sourcing (Krishna et al., 2017) (Gao, Barbier, and Goolsby, 2011), in which the classical question of modeling the accuracies of various labellers without using gold data arises. This work is different in that, not only does it satisfy the conventional crowd sourcing patterns, it also allows for users to describe dependencies between themselves. This work also focuses on being able to label large data using a few functions, an opposite of crowd sourcing settings wherein large number of labellers label small bits of the dataset.

Co-training and Boosting are other well studied procedures similar to Data Programming, but differ in that they don't allow for explicit modelling of dependencies between views of data and that labelled data is explicitly necessary, respectively.

3.3.2 Snorkel Related Work

Combining various sources of weak supervision (Dalvi et al., 2013) (Joglekar, Garcia-Molina, and Parameswaran, 2015) (Zhang et al., 2014) is an important challenge in being able to leverage these sources together and effectively. Researchers have looked at estimating the accuracy of label sources without access to the gold data, mostly set in the crowd sourcing setting, where every user is looked as a source of labels with unknown accuracy. These methods use generative probabilistic models to estimate a latent variable, the true class label, based on noisy observations. Some other methods use user-specified dependency structures to estimate labels. Snorkel is different in that it supports various sources of weak supervision, by learning the correlational structure amongst these sources without ground data.

Snorkel is also related to other forms of supervision such as semi-supervised learning, transfer learning, and active learning, but Snorkel differs in that it's focused on managing weak sources of supervision and doesn't focus on combining itself with other types of supervision.

3.3.3 Babble Labble Related Work

Babble Labble is closely related to natural language explanations/instructions modeling and weak supervision. Most work (Ling and Fidler, 2017) (Liang, Jordan, and Klein, 2013) (Srivastava, Labutov, and Mitchell, 2017) on the natural language processing side, converts these explanations/instructions into features for a discriminative classifier straight away. Babble Labble on the other hand, converts them into labelling functions, sources of weak supervision, which it then uses to label data, and the uses the learnings from the correlation structures of these functions as information whilst building the discriminative classifier.

The related works pertaining to the combination of weak supervision sources is discussed in the above two sub sections.

3.4 A New Paradigm For Data Labelling

3.4.1 Motivation

Machine Learning is bit of a dichotomy in today's world – it's both far more and far less accessible than ever before. On one side, a deep learning model can run state of the art results without any manual feature engineering or algorithm development, thereby making it very accessible. But, there are many ways in which it is opaque, and thought of as black boxes. There aren't ways in which one can specify domain knowledge or heuristics into a model to augment it, and models are often clunky and can't handle a small change in objectives.

However, with the burst of data available, machine learning systems don't need to be hand-programmed at all. These systems simply leverage the data and end up learning the idiosyncrasies and dependencies of the domain from the data. However, this in only true in case of usable data – when it is assembled, clean and debugged, a really expensive and slow task, especially when domain expertise is required. Moreover, in the real world, things are volatile, and the processes might change over time, rendering the old data useless and we'd have to re-iterate the data processing with the updated guidelines. As you can see, this is a horror for scaling.

Hence, for all these reasons, more and more researchers are turning to weaker forms of supervision, such as heuristically generating training data using external knowledge bases, patterns or rules, or other classifiers. These are more or less categorically ways of programmatically generating training data – or, in a catchphrase "Data Programming".

3.4.2 Data Programming

As in many machine learning application settings, the common problems faced are as follows:

- Hand-labelled data is expensive – and procuring it is slow and tedious.
- Lack of external knowledge bases of the domain – rendering traditional distant supervision unusable.
- Constant change in requirements or processes – forcing us to update the models, which isn't an easy task.

Hence, they propose data programming, a paradigm for programmatic creation of training data sets, allowing systems to benefit from the data generated via these systems. In this paradigm, the data is lablled via heuristic rules called labelling functions, rather than labelling each data point by hand.

To write it more formally, a labelling function is $\lambda_i : \chi \mapsto \{-1, 0, 1\}$, a user defined function that encodes domain heuristic and provides a label to some subset of the data. Considering a binary classification task, the objective is to minimize the logistic loss under a linear model given some features,

$$l(w) = \mathbf{E}_{(x,y)\sim\pi}[log(1 + exp(-w^T f(x)y))]$$

where we have some distribution π over object and class pairs $(x, y) \in \chi \times \{-1, 1\}$.

Labelling functions have varying accuracies and coverages, as it represents a pattern that a user wishes to tell their model. It's a simple way of encoding this information than hand-labelling data points, as it quickens up the process. But, in doing so, these functions will tend to overlap, conflict, and have dependencies which users can provide as part of the specification.

Once the user has specified labelling functions, we can first construct a model in which each function behaves independently, given the true class label. If each function λ_i has some probability β_i for labelling an object, and some probability α_i for labelling the object correctly, and assuming the classes have a probability of 0.5 (equal classes), then the model has a distribution of:

$$\mu_{\alpha,\beta}(\Lambda, Y) = 12 \prod_{i=1}^{m} (\beta_i \alpha_i \mathbf{1}_{\{\Lambda_i = Y\}} + \beta_i (1 - \alpha_i) \mathbf{1}_{\{\Lambda_i = -Y\}} + (1 - \beta_i) \mathbf{1}_{\{\Lambda_i = 0\}}) \tag{3.1}$$

where $\Lambda \in \{-1, 0, 1\}^m$ contains the output labels by the labelling functions and $Y \in \{-1, 1\}$ is the predicted class. From here, to find the parameters (α, β) which are most consistent with observations, we simple use maximum likelihood estimation (MLE), which in simple words is maximizing the probability that the observed labels produced on training examples occur under the generative model in (3.1).

From here, once the parameter is learnt, we have to minimize the expected risk over a linear model, so we define the noise aware empirical risk with a regularization term to compute the noise-aware empirical risk minimizer. Since this is a logistic regression problem, it can be solved using stochastic gradient descent as well.

Using empirical evidence from the experiments they conducted, they found that the labelling functions start being dependent on each other and by modelling this structure, they can improve accuracy in some cases. Hence, they extend their model in a way that a user can specify a dependency graph to show how the system can leverage it to better estimate parameters.

3.4.3 Snorkel

Snorkel is the first end-to-end system that implements data programming. It demonstrates that this paradigm enables users to produce high-quality models for a wide range of tasks.

Snorkel's work-flow is designed around data programming and proceeds in three main stages (Figure 3.1):

1. **Writing Labelling Functions:** Users of Snorkel can specify labelling functions, allowing them to express various weak supervision sources such as external knowledge bases, patterns, heuristics, and more.

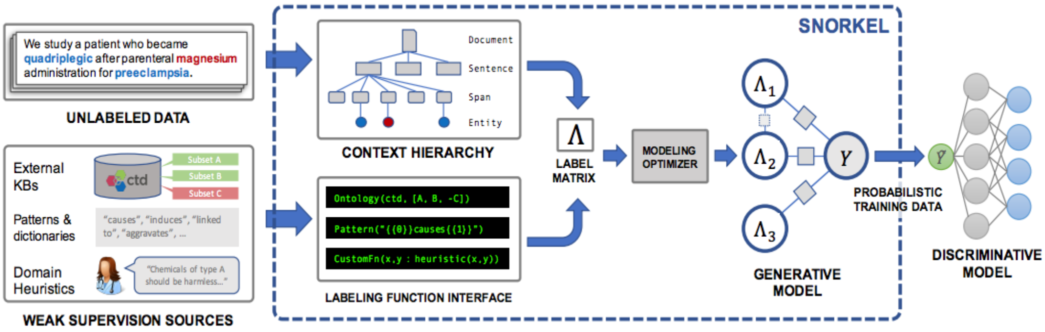

We study a patient who became quadriplegic after parenteral magnesium administration for preeclampsia.
UNLABELED DATA
External KBs
ctd
Subset A
Subset B
Subset C
Patterns & dictionaries
"causes", "induces", "linked to", "aggravates", ...
Domain Heuristics
"Chemicals of type A should be harmless..."
WEAK SUPERVISION SOURCES
SNORKEL
Document
Sentence
Span
Entity
CONTEXT HIERARCHY
Ontology(ctd, [A, B, -C])
Pattern("{{0}}causes{{1}}")
CustomFn(x,y : heuristic(x,y))
LABELING FUNCTION INTERFACE
Λ
LABEL MATRIX
MODELING OPTIMIZER
Λ1
Λ2
Λ3
Y
GENERATIVE MODEL
Ỹ
PROBABILISTIC TRAINING DATA
DISCRIMINATIVE MODEL

FIGURE 3.1: An overview of the Snorkel system

2. **Modelling Accuracies and Correlations:** Then, Snorkel automatically learns a generative model over the functions, to estimate their accuracies and correlations, without using any ground-truth data. It instead learns from the agreements and disagreements of the functions. The generative model is specified as follows:

$$p_w(\Lambda, Y) = Z_w^{-1} exp\left(\sum_{i=1}^{m} w^T \phi_i(\Lambda, y_i)\right)$$

   where Z_w is a normalizing constant, Y are true labels, and ϕ_i are various factors. To learn this model without access to the true labels Y, we can minimize the negative log marginal likelihood given the observed label matrix Λ:

$$\hat{w} = margmin - log \sum_{Y} p_w(\Lambda, Y)$$

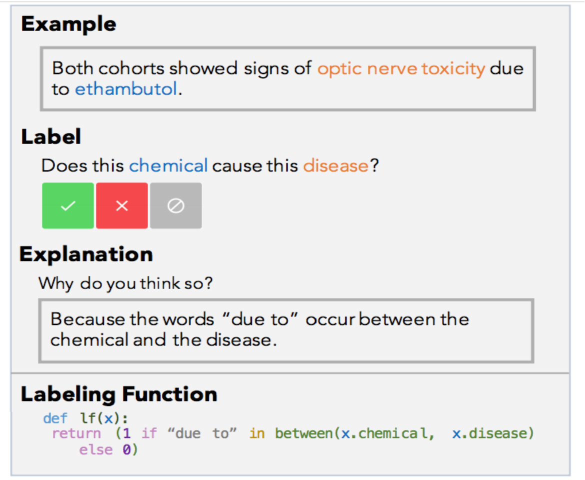
   We can optimize this objective by alternating stochastic gradient with Gibbs's sampling, such as in contrastive divergence.

3. **Training a Discriminative Model:** The output of Snorkel are probabilistic labels, which can be fed into numerous machine learning models (yes, even the discriminative ones now that we have labels). The labels produced by Snorkel are precise but can be low-coverage depending on the functions specified by the user(s), the discriminative model retains this precision but increases coverage and robustness on unseen data.

3.4.4 Babble Labble

Babble Labble is an extension to Snorkel in which users can provide labels as natural language explanations, and hence don't have to specify programmatic functions. This is a big extension, as now this system becomes accessible to people without a programming background, since they can simply write natural language and the system converts these explanations into code (Figure 3.2).

There are three key components to the system: a semantic parser which transforms natural language explanations into a set of logical forms representing labelling functions, a filter bank which removes as many incorrect labelling functions as possible, without needing gold truth labels, and a label aggregator which then combines these labelling functions taking care of overlapping and conflicting functions.



Example

Both cohorts showed signs of optic nerve toxicity due to ethambutol.

Label

Does this chemical cause this disease?

✓ × ⊘

Explanation

Why do you think so?

Because the words "due to" occur between the chemical and the disease.

Labeling Function

```
def lf(x):
 return (1 if "due to" in between(x.chemical, x.disease)
    else 0)
```

FIGURE 3.2: An overview of the Babble Labble system

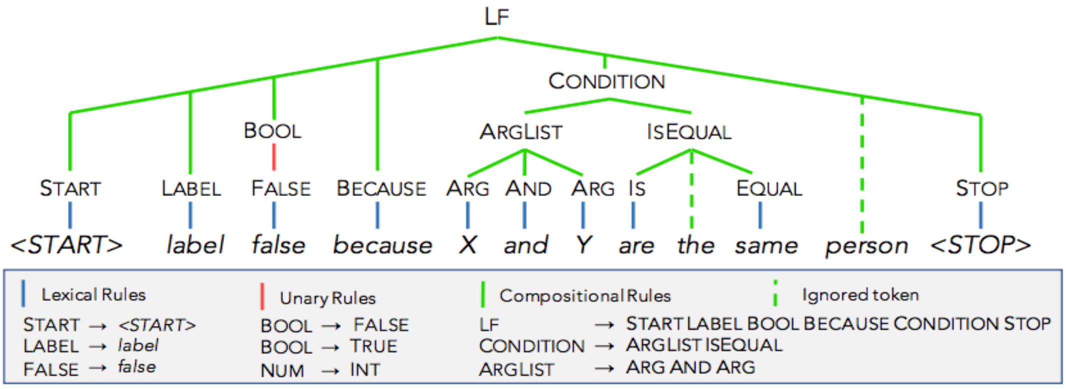


FIGURE 3.3: Rule based semantic parser of the Babble Labble system

Explanations

An user looks at subset S of the unlabelled dataset D, where $|S| \ll |D|$ and for each input $x_i \in S$ provides a label y_i and a natural language explanation e_i for why the example should receive that label. The explanation e_i, usually contains part of the example.

Semantic Parser

The role of the parser is the transform the natural language explanations into labelling functions $\{f_1, \ldots, f_k\}$, of the form $f_i : \chi \mapsto \{-1, 0, 1\}$ in a binary classification setting. The parser doesn't have to be an accurate one that results in the single correct parse of the explanation. However, the main focus if for the parser to have a high coverage, because the hypothesis is that many similar parses can be potentially useful. For this reason, they employ a simple rule based parser, which is usable without any training. They pre-define their set of tokens and predicates for the domain they are working on, based on empirical inputs from subject matter experts. To identify candidate labelling functions, they recursively construct a set of valid parses for each span of the explanation, based on the substitutions in the grammar rules of the rule based parser. They allow any number of tokens in a span to be ignored to match to a rule, as it allows the parser to handle unexpected input and still result in a valid parse. The parser iterates over increasingly large subspans of the input, thereby generating candidates for each entity of the superset of the inputs tokens (Figure 3.3).

Label Aggregator

The label aggregator's role is to combine multiple labels which might be overlapping and conflicting, into one probabilistic label per example. This is already discussed in the previous section describing the functioning of Snorkel.

Later, a discriminative model is trained on top, which maintains the accuracy of the label aggregator, but increases the coverage and makes the system more robust.

3.5 Takeaways

In this section, I've surveyed recent advancements in labelled dataset creation. They allow for the usage of weak supervision sources, in the form of labelling functions which are far more scalable than hand-labelling data points. These are important advancements as we move forward, as data is going to become more and more omnipresent, and being able to use it is literally the fuel of the machine learning industry. Without data, there are no models, and without any models there is no (artificial) intelligence. Thanks to this recent bodies of work, we are starting to understand how we can turn unusable data into something we can leverage to build our models upon.

In the space of online community forums, these techniques lend to be very useful as one can employ them as a help to create datasets for tasks. By using a technique such as Babble Labble, the researchers can also reach to the wider community, most of whom without any knowledge of Machine Learning can help create accurate datasets by using simple natural language.

Chapter 4

Relational Approach To Expert Identification

4.1 Probabilistic Soft Logic

Probabilistic soft logic (PSL) is a framework for probabilistic modeling and collective reasoning in relational domains (Kimmig et al., 2012; Bach et al., 2013). PSL provides a declarative syntax and uses first-order logic to define a templated undirected graphical model over continuous random variables. Like other statistical relational learning methods, dependencies in the domain are captured by constructing rules with weights that can be learned from data. But unlike other statistical relational learning methods, PSL relaxes boolean truth values for atoms in the domain to soft truth values in the interval [0,1]. Triangular norms, which are continuous relaxations of logical connectives AND and OR, are used to combine the atoms in the first-order clauses. In this setting, finding the most probable explanation (MPE), a joint assignment of truth values to all random variable ground atoms, can be done efficiently. As a result of the soft formulation and the triangular norms, the underlying probabilistic model is a hinge-loss Markov random field (HL-MRF) (Bach et al., 2013). Inference in HL-MRFs is a convex optimization problem, which makes working with PSL very efficient in comparison to relational modeling tools that use discrete representations.

For example, a typical PSL rule looks like the following:

$$P(A,B) \bigwedge Q(B,C) \rightarrow R(A,C)$$

where P, Q and R are predicates that represent observed or unobserved attributes in the domain, and A, B, and C are variables. Domain knowledge is captured by writing rules with weights that govern the relative importance of the dependencies between predicates. The groundings of all the rules result in an undirected graphical model that represents the joint probability distribution of assignments for all unobserved atoms, conditioned on the observed atoms.

4.1.1 PSL for Collective Classification

Much research work has focused on computationally modeling online domains as a directed weighted graph. A graph representation of the online domains helps in discovering communities, understanding their structure, seeing how they evolve, and characterizing the interactions of community members.

Given a network and an object o in the network, there are three distinct types of correlations that can be utilized to determine the classification or label of o, (Sen et al., 2008) introduces the idea of collective classification which is the combined classification of a set of interlinked objects using three types of information:

- The correlations between the label of o and the observed attributes of o.
- The correlations between the label of o and the observed attributes (including observed labels) of objects in the neighborhood of o.
- The correlations between the label of o and the unobserved labels of objects in the neighborhood of o.

4.1.2 PSL for Expert Identification

Given the relational nature of online communities, PSL lends itself very nicely to be used for collective reasoning in such settings.

The complicated nature of the community can be represented as first order-like logic rules, and upon inference, the number of groundings of each of those rules will help us in understanding how represented our hypothesis is in the data. The greater the number of groundings, more are the instances of the hypothesis in the data and hence the rule we wrote carries weight. PSL allows for soft logic, and hence when a rule is falsely represented by a data point doesn't break the system. Instead, it simply reduces the weightage attached to the rule. For the purposes of this report, we can assume that each of the rules start with the same weight, and end up with different weights at the end of inference.

4.2 Dataset

In this work, we utilize StackExchange dataset, which has been made available through archive.org, downloaded in Oct, 2017 containing nine years of data since 2008. From this dataset, we specifically work with the StackOverflow website data, which, at that point, the statistics of the dataset are recorded in Table 4.1.

TABLE 4.1: Dataset Stats

Dataset	**#entities**
Badges	24,084,712
Comments	60,098,125
PostHistory	96,412,786
PostLinks	4,406,857
Posts	37,215,528
Tags	50,000
Users	7,617,191
Votes	133,347,053

Some of the fields in the data are as follows:

- **title**: the name given to the question thread

- **original_post**: the body of the question (extends the title)
- **url**: link to the support forum page
- **discussion_id**: unique id for the thread
- **created**: date/time at which the thread was created
- **topic**: community subdivision the question is categorized under
- **asker_id**: user-name of the question seeker
- **user_id**: user-number of the question seeker
- **num_comment**: the number of comments on the thread
- **comments**: list of comments/replies in sequential order
- **comments_progression**: interaction entities within comments
- **correctly_answered**: T/F flagging if a question was marked as resolved
- **correct_answer_index**: index of the correct answer (if any)
- **has_tags**: flag indicating if any candidates for tags were found
- **tags**: candidate tags for the thread (if any). Stored as a list

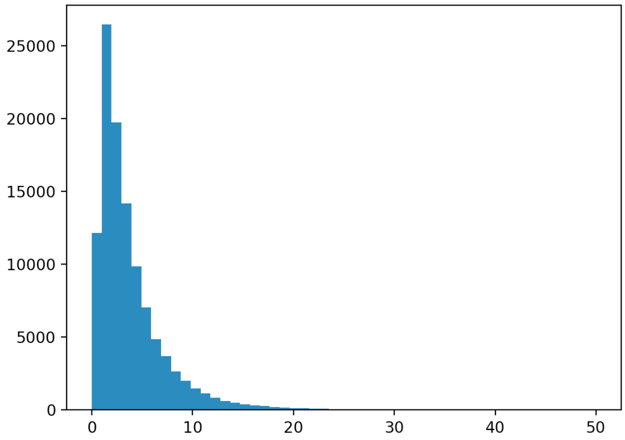


FIGURE 4.1: Distribution of comments vs threads

(Pal, Harper, and Konstan, 2012) add a filter of consider users with more than 10 posts as a way of recording only slightly active users. Then, they consider the top 10% of these based on the number of posts as experts, and the rest 90% as non-experts. (Movshovitz-Attias et al., 2013) filter the data by using the top 1% of users (13087users) that have reputation greater or equal to 2400 and consider them to be the current expert users of StackOverflow. Statistics of this dataset are recorded in Table 4.2.

The relational information, which describes how posts are related to each other is what we use to form our network. There are two kinds of links that are described – “duplicate" and “linked". Source posts are always questions, whereas the target could be an answer/question, but are mostly questions. Duplicate links are when

TABLE 4.2: Dataset Statistics: considering users >10 posts (answers/question)

Dataset	#entities
PostLinks	2,066,870
Posts	29,748,161
Users	516,628

the source and target nodes are marked as duplicate by the user community and linked links are when users mention the target post in the source post's thread. Therefore, if an answer is linked to another answer, the linked link will point from the source question to the target question. There are $\sim 4.1M$ "linked" links and $\sim 570K$ "duplicate" links.

4.3 Model for Expert Identification

The following is the PSL rules used in our model for expert identification in the relational setting.

$$\neg Expert(U)$$

$$a\text{cceptedAnswerOf(P1,P2)} \wedge author(P1,U) \rightarrow expert(U)$$

$$d\text{uplicates(P1,P2)} \wedge author(P1,U) \rightarrow \neg expert(U)$$

$$a\text{nswers(P1,P2)} \wedge answers(P3,P2) \wedge author(P3,U2) \wedge expert(U2) \wedge author(P1,U1) \rightarrow \neg expert(U1)$$

$$a\text{nswers(P1,P2)} \wedge author(P2,U) \rightarrow \neg expert(U)$$

$$d\text{uplicates(P1,P2)} \wedge acceptedAnswerOf(P2,P3) \wedge author(P2,U) \rightarrow expert(U)$$

$$d\text{uplicates(P1,P2)} \wedge acceptedAnswerOf(P3,P2) \wedge author(P3,U) \rightarrow expert(U)$$

Equation 4.3 is the grounding rule, stating that by default no user is an expert and results in 36,293 ground atoms.

Equation 4.3 states that the author of an accepted answer is generally an expert and results in 13,559 ground atoms.

Equation 4.3 says that the author of a post that is marked as a duplicate of another post is generally not an expert, and results in 9218 ground atoms.

Equation 4.3 states that for a question with multiple answers, if an expert has contributed an answer, the authors of the other answers are non-experts and results in 101,726 groundings.

Equation 4.3 says that authors of questions are generally non-experts from the hypothesis that experts tend to answer more questions than ask them and results in 28,990 groundings.

Equation 4.3 states that if a question refers to an accepted answer of a diferent question, then the author of that answer is an expert, and results in 2356 groundings.

Equation 4.3 says that if a question is marked as a duplicate of another question, and the other question has an accepted answer then the author of that answer is an expert and results in 6294 groundings.

4.4 Results

The best results from the (Pal and Konstan, 2010) paper is referred to as M0.

Model	Accuracy	Precision	Recall
M0	0.84	0.87	0.82
Rules 1,2,3	0.77	0.74	0.76
Rules 1,2,3,4,5	0.91	0.89	0.80
Rules 1,2,3,4,5,6,7	**0.94**	**0.91**	**0.86**

Thus, even with some preliminary rules, we see that our model already performs better than the models proposed by (Pal and Konstan, 2010).

4.5 Future Work

The first, and perhaps the most important issue to be tackled is to create a dataset for this task by employing the techniques from Chapter 3. Since using reputation scores for creating the gold labelled data is a weak method, a more robust data creation process is required. More work also needs to be done on the PSL models by including more rules with greater coverages and utilizing more local indicators of expertise, and latent variable modeling.

Bibliography

Adamic, Lada A et al. (2008). “Knowledge sharing and yahoo answers: everyone knows something”. In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 665–674.

Attiaoui, Dorra, Arnaud Martin, and Boutheina Ben Yaghlane (2017). “Belief Measure of Expertise for Experts Detection in Question Answering Communities: case study Stack Overflow”. In: *Procedia Computer Science* 112, pp. 622–631.

Bach, Stephen et al. (2013). “Hinge-loss Markov random fields: Convex inference for structured prediction”. In: *arXiv preprint arXiv:1309.6813*.

Bartle, Richard (1990). “Interactive multi-user computer games”. In: *MUSE Ltd for British Telecom plc, Colchester, UK, Report*.

Brin, Sergey and Lawrence Page (1998). “The anatomy of a large-scale hypertextual web search engine”. In: *Computer networks and ISDN systems* 30.1-7, pp. 107–117.

Craven, Mark, Johan Kumlien, et al. (1999). “Constructing biological knowledge bases by extracting information from text sources.” In: *ISMB*. Vol. 1999, pp. 77–86.

Dalvi, Nilesh et al. (2013). “Aggregating crowdsourced binary ratings”. In: *Proceedings of the 22nd international conference on World Wide Web*. ACM, pp. 285–294.

Gao, Huiji, Geoffrey Barbier, and Rebecca Goolsby (2011). “Harnessing the crowdsourcing power of social media for disaster relief”. In: *IEEE Intelligent Systems* 26.3, pp. 10–14.

Hancock, Braden et al. (2018). “Training Classifiers with Natural Language Explanations”. In: *arXiv preprint arXiv:1805.03818*.

Hsieh, Gary and Scott Counts (2009). “mimir: a market-based real-time question and answer service.” In: *CHI*, pp. 769–778.

Joglekar, Manas, Hector Garcia-Molina, and Aditya Parameswaran (2015). “Comprehensive and reliable crowd assessment algorithms”. In: *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, pp. 195–206.

Kimmig, Angelika et al. (2012). “A short introduction to probabilistic soft logic”. In: *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pp. 1–4.

Kleinberg, Jon M et al. (1999). “The web as a graph: measurements, models, and methods”. In: *International Computing and Combinatorics Conference*. Springer, pp. 1–17.

Krishna, Ranjay et al. (2017). “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International Journal of Computer Vision* 123.1, pp. 32–73.

Liang, Percy, Michael I Jordan, and Dan Klein (2013). “Learning dependency-based compositional semantics”. In: *Computational Linguistics* 39.2, pp. 389–446.

Ling, Huan and Sanja Fidler (2017). “Teaching machines to describe images via natural language feedback”. In: *arXiv preprint arXiv:1706.00130* 2.

Mallory, Emily K et al. (2015). “Large-scale extraction of gene interactions from full-text literature using DeepDive”. In: *Bioinformatics* 32.1, pp. 106–113.

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

Movshovitz-Attias, Dana et al. (2013). "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow". In: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, pp. 886–893.

Pal, Aditya, F Maxwell Harper, and Joseph A Konstan (2012). "Exploring question selection bias to identify experts and potential experts in community question answering". In: *ACM Transactions on Information Systems (TOIS)* 30.2, p. 10.

Pal, Aditya and Joseph A Konstan (2010). "Expert identification in community question answering: exploring question selection bias". In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, pp. 1505–1508.

Ratner, Alexander et al. (2017). "Snorkel: Rapid training data creation with weak supervision". In: *arXiv preprint arXiv:1711.10160*.

Ratner, Alexander J et al. (2016). "Data programming: Creating large training sets, quickly". In: *Advances in Neural Information Processing Systems*, pp. 3567–3575.

Rheingold, Howard (1994). *Building fun online learning communities*.

Sen, Prithviraj et al. (2008). "Collective classification in network data". In: *AI magazine* 29.3, p. 93.

Shah, Chirag and Jefferey Pomerantz (2010). "Evaluating and predicting answer quality in community QA". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 411–418.

Shin, Jaeho et al. (2015). "Incremental knowledge base construction using deepdive". In: *Proceedings of the VLDB Endowment* 8.11, pp. 1310–1321.

Srivastava, Shashank, Igor Labutov, and Tom Mitchell (2017). "Joint concept learning and semantic parsing from natural language explanations". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1527–1536.

Zhang, Yuchen et al. (2014). "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing". In: *Advances in neural information processing systems*, pp. 1260–1268.