Joint Learning with Global Inference for Comment Classification in Community Question Answering

Shafiq Joty, **Lluís Màrquez** and **Preslav Nakov**
ALT Research Group
Qatar Computing Research Institute — HBKU, Qatar Foundation
{sjoty,lmarquez,pnakov}@qf.org.qa

Abstract

This paper addresses the problem of comment classification in community Question Answering. Following the state of the art, we approach the task with a global inference process to exploit the information of all comments in the answer-thread in the form of a fully connected graph. Our contribution comprises two novel joint learning models that are on-line and integrate inference within learning. The first one jointly learns two *node*- and *edge*-level MaxEnt classifiers with stochastic gradient descent and integrates the inference step with loopy belief propagation. The second model is an instance of fully connected pairwise CRFs (FCCRF). The FCCRF model significantly outperforms all other approaches and yields the best results on the task to date. Crucial elements for its success are the global normalization and an Ising-like edge potential.

1 Introduction

Online community fora have been gaining a lot of popularity in recent years. Many of them, such as Stack Exchange[1], are quite open, allowing anybody to ask and anybody to answer a question, which makes them very valuable sources of information. Yet, this same democratic nature resulted in some questions accumulating a large number of answers, many of which are of low quality. While nowadays online fora are typically searched using standard search engines that index entire threads, this is not optimal, as it can be very time-consuming for a user to go through and make sense of a long thread.

Q: *hello guys and gals..could anyone of u knows where to buy a good and originals RC helicopters and toy guns here in qatar..im longin for this toys but its nowhere to find.. thanks*

$\mathbf{A}_1$ Go to Doha city center you may get it at 4 floor.
Local: Good, Human: Good

$\mathbf{A}_2$ "Hobby Shop" in City center has these toys with original motors. They are super cool.. U will love that shop..and will definetly buy one :) Have fun :)
Local: Good, Human: Good

$\mathbf{A}_3$ IM selling all my rc nitro helicopters. call me at 5285113.. (1)TREX 600 new/ (1) TREX500 (1) SHUTTLERG (1) FUTABA ... [truncated]
Local: Good, Human: Bad

$\mathbf{A}_4$ Hobby Shop- City Centre
Local: Bad, Human: Good

$\mathbf{A}_5$ OMG!! :— Guns and helicopters??!!
Local: Good, Human: Bad

$\mathbf{A}_6$ Speed Marine- Salwa Road I think these guys r the best in town...
Local: Good, Human: Good

$\mathbf{A}_7$ City center, i've seen wonderful collection.. Its some wer besides the kids fun place..
Local: Bad, Human: Good

$\mathbf{A}_8$ try the shop in city center. they have many RC toys for sale there. and for the toy guns, in your talking baout airsoft i think its prohibited here. good luck
Local: Good, Human: Good

Figure 1: Example answer-thread with human annotations and automatic predictions by a local classifier at the comment level.

[1] http://stackexchange.com/

Proceedings of NAACL-HLT 2016, pages 703–713,
San Diego, California, June 12-17, 2016.

Thus, the creation of automatic systems for Community Question Answering (cQA), which could provide efficient and effective ways to find good answers in a forum, has received a lot of research attention recently (Duan et al., 2008; Li and Manandhar, 2011; dos Santos et al., 2015; Zhou et al., 2015a; Wang and Ittycheriah, 2015; Tan et al., 2015; Feng et al., 2015; Nicosia et al., 2015; Barrón-Cedeño et al., 2015; Joty et al., 2015). There have been also related shared tasks at SemEval-2015[2] and SemEval-2016[3] (Nakov et al., 2015; Nakov et al., 2016).

In this paper, we focus on the particular problem of classifying comments in the answer-thread of a given question as *good* or *bad* answers. Figure 1 presents an excerpt of a real example from the Qatar-Living dataset from SemEval-2015 Task 3. There is a question on top (Q) followed by eight comments $(A_1, A_2, \cdots, A_8)$. According to the human annotations ('Human'), all comments but 3 and 5 are good answers to Q. The comments also contain the predictions of a *good-vs-bad* binary classifier trained with state-of-the-art features on this dataset (Nicosia et al., 2015); its errors are highlighted in red. Many comments are short, making it difficult for the classifier to make the right decisions, but some errors could be corrected using information in the other comments. For instance, comments 4 and 7 are similar to each other, but also to comments 2 and 8 ('Hobby shop', 'City Center', etc.). It seems reasonable to think that similar comments should have the same labels, so comments 2, 4, 7 and 8 should all be labeled consistently as *good* comments.

Indeed, recent work has shown the benefit of using varied thread-level information for answer classification, either by developing features modeling the thread structure and dialogue (Barrón-Cedeño et al., 2015), or by applying global inference mechanisms at the thread level using the predictions of local classifiers (Joty et al., 2015). We follow the second approach, assuming a graph representation of the answer-thread, where nodes are comments and edges represent pairwise (similarity) relations between them. Classification decisions are at the level of nodes and edges, and global inference is used to get the best label assignment to all comments.

[2]http://alt.qcri.org/semeval2015/task3/

[3]http://alt.qcri.org/semeval2016/task3/

Our main contribution is to propose online models for learning the decisions jointly, incorporating the inference inside the joint learning algorithm. Building on the ideas from papers coupling learning and inference for NLP structure prediction problems (Punyakanok et al., 2005; Carreras et al., 2005), we propose joint learning of two MaxEnt classifiers with stochastic gradient descent, integrating global inference based on loopy belief propagation. We also propose a joint model with global normalization, that is an instance of Fully Connected Conditional Random Fields (Murphy, 2012). We compare our joint models with the previous state of the art for the comment classification problem. We find that the coupled learning-and-inference model is not competitive, probably due to the label bias problem. On the contrary, the fully connected CRF model improves results significantly over all rivaling models, yielding the best results on the task to date.

In the remainder of this paper, after discussing related work in Section 2, we introduce our joint models in Section 3. We then describe our experimental settings in Section 4. The experiments and analysis of results are presented in Section 5. Finally, we summarize our contributions with future directions in Section 6.

2 Related Work

The idea of using global inference based on locally learned classifiers has been tried in various settings. In the family of graph-based inference, Pang and Lee (2004) used local classification scores with proximity information as edge weights in a graph-cut inference to collectively identify subjective sentences in a review. Thomas et al. (2006) used the same framework to classify congressional transcribed speeches. They applied a classifier to provide edge weights that reflect the degree of agreement between speakers. Burfoot et al. (2011) extended the framework by including other inference algorithms such as loopy belief propagation and mean-field.

"Learning and inference under structural and linguistic constraints" (Roth and Yih, 2004) is a framework to combine the predictions of local classifiers in a global decision process solved by Integer Linear Programming.

The framework has been applied to many NLP structure prediction problems, including shallow parsing (Punyakanok and Roth, 2000), semantic role labeling (Punyakanok et al., 2004), and joint learning of entities and relations (Roth and Yih, 2004). Further work explored the possibility of coupling learning and inference in the previous setting. For instance, Carreras et al. (2005) presented a model for parsing that jointly trains several local decisions with a perceptron-like algorithm that gets feedback after inference. Punyakanok et al. (2005) studied empirically and theoretically the cases in which this *inference-based learning* strategy is superior to the decoupled approach.

On the particular problem of comment classification in cQA, we find some work exploiting thread-level information. Hou et al. (2015) used features about the position of the comment in the thread. Barrón-Cedeño et al. (2015) developed more elaborated global features to model thread structure and the interaction among users. Other work exploited global inference algorithms at the thread-level. For instance, (Zhou et al., 2015c; Zhou et al., 2015b; Barrón-Cedeño et al., 2015) treated the task as sequential classification, using a variety of machine learning algorithms to label the sequence of time-sorted comments: LSTMs, CRFs, SVM^{hmm}, etc. Finally, Joty et al. (2015) showed that exploiting the pairwise relations between comments (at any distance) is more effective than the sequential information. Their results are the best on this task to date. In this paper, we assume the same setting (cf. Section 3) and we experiment with new models to do learning jointly with inference in the same manner as in (Punyakanok et al., 2005), and also using fully connected pairwise CRFs.

3 Our Model

Given a forum question Q and a thread of answers $T = \{A_1, A_2, \cdots, A_n\}$, the task is to classify each answer A_i in the thread into one of K possible classes based on its relevance to the question. We represent each thread as a fully-connected graph, where each node represents an answer in the thread.

Given this setting, there exist at least three fundamentally different approaches to learn classification functions.

First, the traditional approach of learning a local classifier ignoring the structure in the output and using it to predict the label of each node A_i separately. This approach only considers correlations between the label of A_i and features extracted from A_i.

The second approach, adopted by Joty et al. (2015), is to first learn two local classifiers separately: (*i*) a node-level classifier to predict the label for each individual node, and (*ii*) an edge-level classifier to predict whether the two nodes connected by an edge should have the same label or not (assuming a fully connected graph). The predictions of the local classifiers are then used in a global inference algorithm (e.g., graph-cut) to perform collective classification by maintaining structural constraints in the output. There are two issues with this model: (*i*) the local classifiers are trained separately; (*ii*) by decoupling learning from inference, this approach can lead to suboptimal solutions, as Punyakanok et al. (2005) pointed out.

The third approach, which we adopt in this paper, is to model the dependencies between the output variables while learning the classification functions jointly by optimizing a global performance criterion. The dependencies are captured using node-level and edge-level factors defined over a fully connected graph. The idea is that incorporating structural constraints in the form of all-pair relations during training can yield a better solution that directly optimizes an objective function for the target task.

Before we present our models in subsections 3.1 and 3.2, let us first introduce the notation that we will use. Each thread $T = \{A_1, A_2, \cdots, A_n\}$ is represented by a complete graph $G = (V, E)$. Each node $i \in V$ in the graph is associated with an input vector $\mathbf{x}_i$, which represents the features of an answer A_i, and an output variable $y_i \in \{1, 2, \cdots, K\}$, representing the class label. Similarly, each edge $(i, j) \in E$ is associated with an input feature vector $\phi(\mathbf{x}_i, \mathbf{x}_j)$, derived from the node-level features, and an output variable $y_{i,j} \in \{1, 2, \cdots, L\}$, representing the labels for the pair of nodes. We use $\psi_n(y_i|\mathbf{x}_i, \mathbf{v})$ and $\psi_e(y_{i,j}|\phi(\mathbf{x}_i, \mathbf{x}_j), \mathbf{w})$ to denote the node-level and the edge-level classification functions, respectively. We call ψ_n and ψ_e factors, which can be either normalized (e.g., probabilities) or unnormalized quantities. The model parameters $\theta = [\mathbf{v}, \mathbf{w}]$ are to be learned during training.

3.1 Joint Learning of Two Classifiers with Global Thread-Level Inference

Our aim is to train the local classifiers so that they produce correct global classification. To this end, in our first model we train the node- and the edge-level classifiers jointly based on global feedback provided by a global inference algorithm. The global feedback determines how much to adjust the local classifiers so that the classifiers and the inference together produce the desired result. We use log-linear models (aka maximum entropy) for both classifiers:

$$\psi_n(y_i = k|\mathbf{x_i}, \mathbf{v}) = \frac{\exp(\mathbf{v}_k^T \mathbf{x_i})}{\sum_{k'=1}^{K} \exp(\mathbf{v}_k^T \mathbf{x_i})} \quad (1)$$

$$\psi_e(y_{i,j} = l|\phi(\mathbf{x_i}, \mathbf{x_j}), \mathbf{w}) = \frac{\exp(\mathbf{w}_l^T \phi(\mathbf{x_i}, \mathbf{x_j}))}{\sum_{l'=1}^{L} \exp(\mathbf{w}_{l'}^T \phi(\mathbf{x_i}, \mathbf{x_j}))} \quad (2)$$

The log likelihood (LL) for one data point $(\mathbf{x}, \mathbf{y})$ (i.e., a thread) can be written as follows:

$$f(\theta) = \sum_{i \in V} \sum_{k=1}^{K} y_i^k \left[\mathbf{v}_k^T \mathbf{x_i} - \log Z(\mathbf{v}, \mathbf{x_i})\right] + \sum_{(i,j) \in E} \sum_{l=1}^{L} y_{i,j}^l \left[\mathbf{w}_l^T \phi(\mathbf{x_i}, \mathbf{x_j}) - \log Z(\mathbf{w}, \mathbf{x_i}, \mathbf{x_j})\right] \quad (3)$$

where y_i^k and $y_{i,j}^l$ are the gold labels for i-th node and (i,j)-th edge expressed in 1-of-K (or 1-of-L) encoding, respectively, and $Z(\cdot)$ terms are the local normalization constants.

We give a pseudocode in Algorithm 1 that trains this model in an online fashion using feedback from the loopy belief propagation (LBP) inference algorithm (to be described later in Section 3.1.1). Specifically, the marginals from the LBP are used in a stochastic gradient descent (SGD) algorithm, which has the following (minibatch) update rule:

$$\theta_{t+1} = \theta_t - \eta_t \frac{1}{N} f'(\theta_t) \quad (4)$$

where θ_t and η_t are the model parameters and the learning rate at step t, respectively, and $\frac{1}{N} f'(\theta_t)$ is the mean gradient for the minibatch (a thread). For our maximum entropy models, the gradients become

$$f'(\mathbf{v}) = \sum_{i \in V} \left[\beta_n(y_i) - y_i\right] . \mathbf{x}_i \quad (5)$$

$$f'(\mathbf{w}) = \sum_{(i,j) \in E} \left[\beta_e(y_{i,j}) - y_{i,j}\right] . \phi(\mathbf{x_i}, \mathbf{x_j}) \quad (6)$$

Algorithm 1: Joint learning of local classifiers with global thread-level inference

1. Initialize the model parameters $\mathbf{v}$ and $\mathbf{w}$;
2. **repeat**
   - **for** *each thread* $G = (V, E)$ **do**
     - a. Compute node and edge probabilities $\psi_n(y_i|\mathbf{x}_i, \mathbf{v})$ and $\psi_e(y_{i,j}|\phi(\mathbf{x}_i, \mathbf{x}_j), \mathbf{w})$;
     - b. Infer node and edge marginals $\beta_n(y_i)$ and $\beta_e(y_{i,j})$ using sum-product LBP;
     - c. Update: $\mathbf{v} = \mathbf{v} - \frac{\eta}{|V|} f'(\mathbf{v})$;
     - d. Update: $\mathbf{w} = \mathbf{w} - \frac{\eta}{|E|} f'(\mathbf{w})$;
   - **end**
   - **until** *convergence*;

In the above equations, β and y are the marginals and the gold labels, respectively.

Note that when applying the model to the test threads, we need to perform the same global inference to get the best label assignments.

3.1.1 Inference Using Belief Propagation

Belief Propagation or BP (Pearl, 1988) is a message passing algorithm for inference in probabilistic graphical models. It supports (*i*) *sum-product,* to compute the marginal distribution for each unobserved variable, i.e., $p(y_i|\mathbf{x}, \theta)$; and (*ii*) *max-product,* to compute the most likely label configuration, i.e., $\text{argmax}_{\mathbf{y}}\, p(\mathbf{y}|\mathbf{x}, \theta)$. We describe here the variant that operates on undirected graphs (aka Markov random fields) with pairwise factors, which uses the following equations:

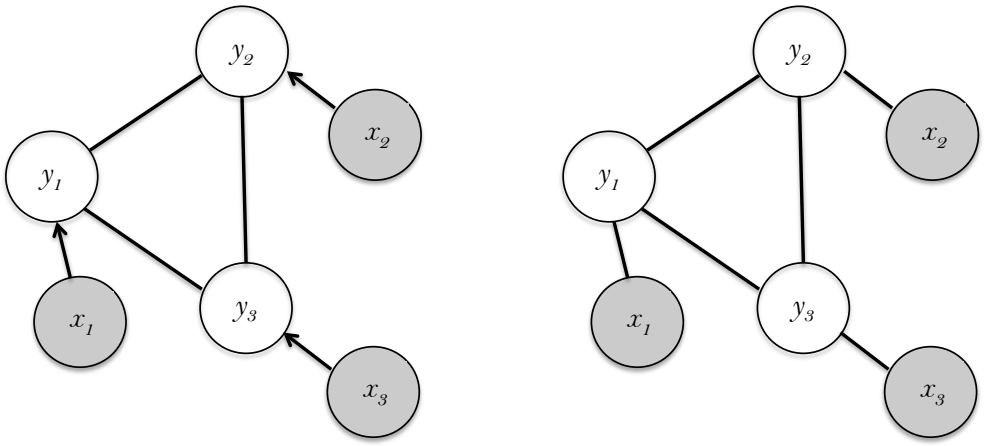
$$\mu_{i \to j}(y_j) = \sum_{y_i} \psi_n(y_i) \psi_e(y_{i,j}) \prod_{k \in N(i) \setminus j} \mu_{k \to i}(y_i) \quad (7)$$

$$\beta_n(y_i) \approx \psi_n(y_i) \prod_{j \in N(i)} \mu_{j \to i}(y_i) \quad (8)$$

where $\mu_{i \to j}$ is a message from node i to node j, $N(i)$ are the nodes neighbouring i, and $\psi_n(y_i)$ and $\psi_e(y_{i,j})$ are the node and the edge factors.

The algorithm proceeds by sending messages on each edge until the node beliefs $\beta_n(y_i)$ stabilize. The edge beliefs can be written as follows:

$$\beta_e(y_{i,j}) \approx \psi_e(y_{i,j}) \times \mu_{i \to j}(y_i) \times \mu_{j \to i}(y_j) \quad (9)$$



(a) Locally normalized joint model

(b) Globally normalized joint model

Figure 2: Graphical representation of our two joint models: (a) a joint model with locally normalized factors; (b) a joint model with global normalization, i.e., a fully connected conditional random field.

The node and the edge marginals are then computed by normalizing the node and the edge beliefs, respectively. By replacing the summation with a $\max$ operation in Equation 7, we can get the most likely label configuration (i.e., argmax over labels).

BP is guaranteed to converge to an exact solution if the graph is a tree. However, exact inference is intractable for general graphs, i.e., graphs with loops. Despite this, it has been advocated by Pearl (1988) to use BP in loopy graphs as an approximation scheme; see also (Murphy, 2012), page 768. The algorithm is then called "loopy" BP, or LBP. Although LBP gives approximate solutions for general graphs, it often works well in practice (Murphy et al., 1999), outperforming other methods such as mean field (Weiss, 2001) and graph-cut (Burfoot et al., 2011).

It is important to mention that the approach presented above (i.e., subsection 3.1) is similar in spirit to the approach of Collins (2002), Carreras and Màrquez (2003) and Punyakanok et al. (2005). The main difference is that they use a Perceptron-like online algorithm, where the updates are done based on the best label configuration (i.e., $\operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \theta)$) rather than the marginals.

One can use graph-cut (applicable only for binary output variables) or max-product LBP for the decoding task. However, this yields a discontinuous estimate (even with averaged perceptron) for the gradient (see Section 5). For the same reason, we use sum-product LBP rather than max-product LBP.

3.2 A Joint Model with Global Normalization

Although the approach of updating the parameters of the local classifiers based on the global inference might seem like a natural extension to train the classifiers jointly, it suffers from at least two problems. First, since the node and the edge scores are normalized locally (see Equations 1 and 2), this approach leads to the so-called *label bias* problem, previously discussed by Lafferty et al. (2001). Namely, due to the local normalization, local features at any node do not influence states of other nodes in the graph. Second, the two classifiers use their own feature sets. However, the same feature sets that give optimal results locally (i.e., when trained on local objectives), may not work well when the models are trained jointly based on the global feedback. In order to address these issues, below we propose a different model.

In our second approach, we seek to build a joint model with global normalization. We define the following conditional joint distribution:

$$p(\mathbf{y}|\mathbf{v}, \mathbf{w}, \mathbf{x}) = \frac{1}{Z(\mathbf{v}, \mathbf{w}, \mathbf{x})} \prod_{i \in V} \psi_n(y_i|\mathbf{x}, \mathbf{v}) \cdot \prod_{(i,j) \in E} \psi_e(y_{i,j}|\mathbf{x}, \mathbf{w}) \quad (10)$$

where ψ_n and ψ_e are the node and edge *factors*, and $Z(\cdot)$ is the global normalization constant that ensures a valid probability distribution.

This model is essentially a fully connected conditional random field or FCCRF (Murphy, 2012). Figure 2 shows the differences between the two models with the standard graphical model representation.[4] The global normalization allows CRFs to take long-range interactions into account. Similar to our previous model, we use a log-linear representation for the factors:

$$\psi_n(y_i|\mathbf{x},\mathbf{v}) = \exp(\mathbf{v}^T\phi(y_i,\mathbf{x})) \quad (11)$$

$$\psi_e(y_{i,j}|\mathbf{x},\mathbf{w}) = \exp(\mathbf{w}^T\phi(y_{i,j},\mathbf{x})) \quad (12)$$

where $\phi(\cdot)$ is a feature vector derived from the inputs and the labels. The LL for one data point becomes

$$f(\theta) = \sum_{i\in V}\mathbf{v}^T\phi(y_i,\mathbf{x}) + \sum_{(i,j)\in E}\mathbf{w}^T\phi(y_{i,j},\mathbf{x}) - \log Z(\mathbf{v},\mathbf{w},\mathbf{x}) \quad (13)$$

This objective is convex, so we can use gradient-based methods to find the global optimum. The gradients have the following form:

$$f'(\mathbf{v}) = \sum_{i\in V}\phi(y_i,\mathbf{x}) - \mathbb{E}[\phi(y_i,\mathbf{x})] \quad (14)$$

$$f'(\mathbf{w}) = \sum_{(i,j)\in E}\phi(y_{i,j},\mathbf{x}) - \mathbb{E}[\phi(y_{i,j},\mathbf{x})] \quad (15)$$

where $\mathbb{E}[\phi(.)]$ terms denote the expected feature vector. Traditionally, CRFs have been trained using offline methods like limited-memory BFGS. Online training of CRFs using SGD was proposed by Vishwanathan et al. (2006). To compare our two methods, we use SGD to train our CRF models. The pseudocode is very similar to Algorithm 1.

3.2.1 Modeling Edge Factors

One crucial aspect in the joint models described above is the modeling of edge factors. The traditional way is to define edge factors, where $y_{i,j}$ spans over all possible state transitions, that is K^2 different transitions, each of which is associated with a weight vector. This method has the advantage that it models transitions in a fine-grained way, but, in doing so, it also increases the number of model parameters, which may result in overfitting.

[4]Edge level features and output variables are not shown in Figure 2 to avoid visual clutter.

Alternatively, one can define *Ising-like* edge factors, where we only distinguish between two transitions: (*i*) *same*, when $y_i = y_j$ and (*ii*) *different*, when $y_i \neq y_j$. This modeling involves tying one set of parameters for all *same* transitions, and another set for all *different* transitions.

4 Experimental Setting

In this section, we describe our experimental setting. We first introduce the dataset we use, then we present the features and the models that we compare.

4.1 Datasets and Evaluation

We experimented with the dataset from SemEval-2015 Task 3 on Answer Selection for Community Question Answering (Nakov et al., 2015). The dataset contains question-answer threads from the Qatar Living forum.[5] Each thread consists of a question followed by one or more (up to 143) comments. The dataset is split into training, development and test sets, with 2,600, 300, and 329 questions, and 16,541, 1,645, and 1,976 answers, respectively.

Each comment in the dataset is annotated with one of the following labels, reflecting how well it answers the question: *Good*, *Potential*, *Bad*, *Dialogue*, *Not English*, and *Other*. At SemEval-2015 Task 3, the latter four classes were merged into *BAD* at testing time, and the evaluation measure uses a macro-averaged F_1 over the three classes: *Good*, *Potential*, and *BAD*. Unfortunately, the *Potential* class was both the smallest (covering about 10% of the data), and also the noisiest and the hardest to predict; yet, its impact was magnified by the macro-averaged F_1. Thus, subsequent work has further merged *Potential* under *BAD* (Barrón-Cedeño et al., 2015; Joty et al., 2015), and has used for evaluation F_1 with respect to the *Good* category (or just accuracy). For our experiments below, we also report F_1 for the *Good* class and the overall accuracy. We further perform statistical significance tests using an approximate randomization test based on accuracy.[6] We used SIGF V.2 (Padó, 2006) with 10,000 iterations.

[5]http://www.qatarliving.com/forum

[6]Significance tests operate on individual instances rather than individual classes; thus, they are not applicable for F_1.

4.2 Features

For comparison, we use the features from our previous work (Joty et al., 2015) to implement all classifiers in our models and baselines. There are two sets of features, corresponding to the two main classification problems in the models: node-level (i.e., classifying a comment as *good* or *bad*) and edge-level (i.e., classifying a pair of comments as having the *same* or *different* labels).

The features for node-level classification include three types of information: (*i*) a variety of textual similarity measures computed between the question and the comment, (*ii*) several boolean features capturing the presence of certain relevant words or patterns, e.g., URLs, emails, positive/negative words, acknowledgements, forum categories, presence of long words, etc., and (*iii*) a set of global features modeling dialogue and user interactions in the answer-thread. The features in the last two categories are manually engineered (Nicosia et al., 2015; Barrón-Cedeño et al., 2015).

The features we use for edge-level classification include (*i*) all features from the node classification problem coded as the absolute value of the difference between the two comments, (*ii*) a variety of text similarity features between the two comments, (*iii*) the *good*/*bad* predictions of the node-level classifier on the two comments involved in the edge decision. See (Joty et al., 2015) and (Barrón-Cedeño et al., 2015) for a detailed description of the features.

4.3 Methods Compared

We experimentally compare our above-described joint models to some baselines and to the state of the art for this problem. We briefly describe all models below, together with the names used in the tables of results.

Independent Comment Classification (ICC) These are binary classifiers to label thread comments independently into *good* and *bad* categories. The simplest baseline (Majority) classifies all examples with the most frequent category. We also train a MaxEnt classifier with stochastic gradient descent (SGD) and a voted perceptron (ICC_{ME} and ICC_{Perc}, respectively).

Learning-and-Inference Models (LI) This is the approach presented by Joty et al. (2015), who report the best results on the task. The model is explained in Section 3. We experiment with MaxEnt classifiers trained on-line with SGD and two different inference strategies, graph-cut and loopy BP (LI_{ME-GC} and LI_{ME-LBP}, in our notation).

Joint Learning and Inference Models These are our new models. First, we experiment with the model for joint learning of two classifiers coupled with thread-level inference (Section 3.1). We have two versions, one using MaxEnt classifiers and the other using averaged Perceptron. The inference algorithm is loopy BP in both cases. We call these methods Joint_{ME-LBP} and $\text{Joint}_{Perc-LBP}$, respectively. Second, we experiment with the joint model with global normalization (cf. Section 3.2). We call it FCCRF, for fully connected CRF. We use the *Ising-like* edge factors defined in Section 3.2.1.

5 Evaluation

All results we report below are calculated in the test set, using parameters tuned on the development set.

Our main results are shown in Table 1, where we report accuracy (Acc) as well as precision (P), recall (R) and F_1-score (F_1) for the *good* class.

The models are organized in four blocks. On top, we see that the majority class baseline achieves accuracy of 50.5%, as the dataset is very well balanced between the classes.

In block II, we find the results for the local classifiers, ICC_{ME} and ICC_{Perc}, which achieve very similar results. They are comparable to MaxEnt in Table 2, where we report the best published results on this dataset; yet, our classifiers are trained on-line.

Block III in the table reports results for models that train two local MaxEnt classifiers and then perform thread-level inference using either graph-cut (LI_{ME-GC}) or loopy BP (LI_{ME-LBP}).[7] This yields improvements over the ICC models with the thread-level inference in block II, which is consistent with the findings of (Joty et al., 2015); however, the difference in terms of accuracy is not statistically significant (p-value = 0.09).

[7] Given that MaxEnt and Perceptron perform comparably in this setting, we have just used MaxEnt as it provides directly the class probabilities needed for the thread-level inference.

	Model	*Learning*	*Inference*	P	R	F_1	Acc
I.	Majority	–	–	50.5	100.0	67.1	50.5
II.	ICC_{ME}	Local, SGD	–	75.1	85.8	80.1	78.5
	ICC_{Perc}	Local, Voted	–	76.6	82.4	79.4	78.4
III.	LI_{ME-GC}	Local, SGD	Graph-cut	**77.4**	83.6	80.4	79.4
	LI_{ME-LBP}	Local, SGD	LBP	76.4	84.6	80.3	79.1
IV.	$Joint_{ME-LBP}$	2 classifiers, Joint, SGD	LBP	76.1	84.4	80.0	78.7
	$Joint_{Perc-LBP}$	2 classifiers, Joint, AVG	LBP	77.1	74.5	75.8	76.0
	FCCRF	Joint, SGD	LBP	77.3	**86.2**	**81.5**	**80.5**

Table 1: Results of all compared models on the test set. The best results are boldfaced.

Model	P	R	F_1	Acc
MaxEnt classifier	75.7	84.3	79.8	78.4
Linear CRF	74.9	83.5	78.9	77.5
MaxEnt+ILP	77.0	83.5	80.2	79.1
MaxEnt+GraphCut	**78.3**	82.9	80.6	79.8
Our method (FCCRF)	77.3	**86.2**	**81.5**	**80.5**

Table 2: Comparison to the best published results on the same datasets, as reported in (Joty et al., 2015).

Comparing our LI_{ME-GC} to MaxEnt+GraphCut in Table 2, we see that we are slightly worse: -0.2 in F_1-score, and -0.4 in accuracy. It turns out that this is due to our on-line MaxEnt classifier for the pairwise classification being slightly worse (-0.4 accuracy points absolute), which could explain the lower performance after the graph-cut inference.

Next, block IV shows that the fully connected CRF model (FCCRF) improves over the models in block III by more than one point absolute in both F_1 and accuracy. The improvement is statistically significant (p-value = 0.04); especially noticeable is the increase in recall (+2.6 points). This result is also an improvement over the state of the art, as Table 2 shows.

Again in block IV, we can see that the two models that perform joint training of two classifiers and then integrate inference in the training loop, $Joint_{ME-LBP}$ and $Joint_{Perc-LBP}$, do not work well and fall below the learning and inference models from block III. As we explained above, these models have two major disadvantages compared to FCCRF: (*i*) the local normalization of node and edge scores is prone to label bias issues; (*ii*) each of the two classifiers uses its own feature set, which might not be optimal when they are trained jointly based on the global feedback.

Notice that the version using Perceptron, $Joint_{Perc-LBP}$, works bad in this setting. Since updates are done after each thread-level inference, we could not use a voted perceptron, but an averaged one (Collins, 2002). Moreover, it did not yield probabilities but real-valued scores, which we had to remap to the [0;1] interval using a sigmoid.

5.1 CRF Variants Analysis

Table 3 compares different variants of CRF. The first two rows show the results for the commonly used linear-chain CRF (LCCRF) of order 1 and 2. We can see that these models fall two accuracy (and F_1) points below FCCRF, which indicates that the pairwise relations between non-consecutive comments provide additional relevant information for the task. The fourth row shows the results when we eliminate the edge-level features and we consider state transitions using the bias features only: the decrease in performance is tiny, which means that what matters is to model the interaction in the first place; the particular features used are less important. More noticeable is the effect of using Ising-like modeling of the edge factors in our FCCRF model. If we use finer-grained edge factors for each of the four combinations (Good-Good, Good-Bad, Bad-Good, and Bad-Bad), the performance decreases significantly, mostly due to a drop in recall (see 'FCCRF (4C)').

5.2 Error Analysis

Next, we get a closer look at the predictions made by our best Local (ICC_{ME}), Inference (LI_{ME-GC}), and Global (FCCRF) models. We focus on questions for which there are at least two comments. There were 280 such test questions (out of 329), with a total of 1,927 comments.

Model	P	R	F_1	Acc
LCCRF (ord=1)	76.1	83.2	79.4	78.3
LCCRF (ord=2)	76.8	82.1	79.3	78.4
FCCRF	77.3	**86.2**	**81.5**	**80.5**
FCCRF-noFeatures	77.2	86.0	81.4	80.1
FCCRF (4C)	**78.8**	79.7	79.3	79.0

Table 3: Results for different variants of the joint CRF model on the test set.

The Local, the Inference, and the Joint models made correct predictions for 78.7%, 79.1% and 80.4% of the comments, respectively. We can see that the Inference model behaves more like Local, and not so much like Joint. This is indeed further confirmed when we look at the agreement between each pair of models: Local vs. Inference has 6.0% disagreement, for Local vs. Joint it is 9.9%, and for Inference vs. Joint it is 8.8%.

Figure 3 compares the three models vs. the gold human labels on a particular test question (ID=Q2908; some long comments are truncated and the four omitted answers were classified correctly by all three classifiers). We can see that the Joint model is more robust than the Local one: while Joint corrects two of the three wrong classifications of Local, Inference makes two further errors instead.

Q: *I have a female friend who is leaving for a teaching job in Qatar in January. What would be a useful portable gift to give her to take with her?*

$\mathbf{A}_1$ A couple of good best-selling novels. [. . .]
Loc: Good, Inf: Good, Jnt: Good, Hum: Good

$\mathbf{A}_5$ A big box of decent tea.... like "Scottish blend" or "Tetleys".. [. . .]
Loc: Good, Inf: Good, Jnt: Good, Hum: Good

$\mathbf{A}_6$ Bacon. Nice bread, bacon, bacon, errmmm bacon and a pork joint..
Loc: Good, Inf: Bad, Jnt: Good, Hum: Good

$\mathbf{A}_8$ Go to Tesco buy some good latest DVD.. [. . .]
Loc: Good, Inf: Good, Jnt: Good, Hum: Good

$\mathbf{A}_9$ Couple of good novels, All time favorite movies, ..
Loc: Good, Inf: Bad, Jnt: Good, Hum: Good

$\mathbf{A}_{10}$ Agree I do the same Indorachel..But some time you get a good copy some time a bad one.. [. . .]
Loc: Good, Inf: Good, Jnt: Good, Hum: Bad

$\mathbf{A}_{11}$ Ditto on the books and dvd's. Excedrin.
Loc: Bad, Inf: Bad, Jnt: Good, Hum: Good

$\mathbf{A}_{12}$ Ditto on the bacon, pork sausage, pork chops, ham,..can you tell we miss pork! [. . .]
Loc: Bad, Inf: Bad, Jnt: Good, Hum: Good

Figure 3: Sample test question with a thread of comments and, for each comment, decisions by the local (Loc), the global inference (Inf), and the global joint (Jnt) classifiers, as well as by the human annotators.

6 Conclusion

We have proposed two learning methods for comment classification in community Question Answering. We depart from the state-of-the-art knowledge that exploiting the interrelations between all the comments in the answer-thread is beneficial for the task. Thus, we take as our baseline the learning and inference model from Joty et al. (2015), in which the answer-thread is modeled as a fully connected graph. Our contribution consists of moving the framework to on-line learning and proposing two models for coupling learning with inference.

Our first model learns jointly the two MaxEnt classifiers with SGD and incorporates the graph inference at every step with loopy belief propagation. This model, due to its local normalization, suffers from the label bias problem. The alternative we proposed is to use an instance of a Fully Connected CRF that operates on the same graph and considers the node and edge factors with a shared set of features. One of the main advantages is that the normalization is global. We experimented with the SemEval-2015 Task 3 dataset and we confirmed the advantage of the FCCRF model, which outperforms all baselines and achieves better results than the state of the art.

In the near future, we plan to apply the FCCRF model to the full cQA task, i.e., finding good answers to newly-asked questions using previously-asked questions and their answer threads. In this setting, we want to experiment with (*i*) ranking comments (instead of classifying them), (*ii*) exploiting the similarities between the new question and the questions in the database and also the relations between comments across different answer-threads.

Acknowledgments

This research was performed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), HBKU, part of Qatar Foundation. It is part of the Interactive sYstems for Answer Search (IYAS) project, which is developed in collaboration with MIT-CSAIL.

References

Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, pages 687–693, Beijing, China.

Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective classification of congressional floor-debate transcripts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1506–1515, Portland, Oregon.

Xavier Carreras and Lluís Màrquez. 2003. Online learning via global feedback for phrase recognition. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems*, NIPS '03, Whistler, Canada. MIT Press.

Xavier Carreras, Lluís Màrquez, and Jorge Castro. 2005. Filtering–ranking perceptron learning for partial parsing. *Machine Learning*, 60:41–71.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 1–8, Philadelphia, Pennsylvania, USA.

Cicero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, ACL-IJCNLP '15, pages 694–699, Beijing, China.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics and the Human Language Technology Conference*, ACL-HLT '08, pages 156–164, Columbus, Ohio, USA.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding*, ASRU '15, pages 813–820, Scottsdale, Arizona, USA.

Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. 2015. HITSZ-ICRC: Exploiting classification approach for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 196–202, Denver, Colorado, USA.

Shafiq Joty, Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 573–578, Lisbon, Portugal.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, California, USA.

Shuguang Li and Suresh Manandhar. 2011. Improving question recommendation by exploiting information need. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ACL '11, pages 1425–1434, Portland, Oregon, USA.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 467–475, Stockholm, Sweden.

Kevin Murphy. 2012. *Machine Learning A Probabilistic Perspective*. The MIT Press.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 269–281, Denver, Colorado, USA.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeño, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, Lluís Màrquez, Shafiq Joty, and Walid Magdy. 2015. QCRI: Answer selection for community question answering - experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 203–209, Denver, Colorado, USA.

Sebastian Padó, 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, pages 271–278, Barcelona, Spain.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, California, USA.

Vasin Punyakanok and Dan Roth. 2000. Shallow parsing by inferencing with classifiers. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*, ConLL '00, pages 107–110, Lisbon, Portugal.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Geneva, Switzerland.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2005. Learning and inference over constrained output. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI' 05, pages 1124–1129, Edinburgh, Scotland.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning*, CoNLL '04, pages 1–8, Boston, Massachusetts, USA.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 327–335, Sydney, Australia.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. 2006. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 969–976, Pittsburgh, Pennsylvania, USA.

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *arXiv preprint arXiv:1507.02628*.

Yair Weiss. 2001. Comparing the mean field method and belief propagation for approximate inference in MRFs. *Advanced Mean Field Methods*.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015a. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL-IJCNLP '15, pages 250–259, Beijing, China.

Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, Buzhou Tang, and Xiaolong Wang. 2015b. Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 713–718, Beijing, China.

Xiaoqiang Zhou, Baotian Hu, Jiaxin Lin, Yang Xiang, and Xiaolong Wang. 2015c. ICRC-HIT: A deep learning based comment sequence labeling system for answer selection challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 210–214, Denver, Colorado, USA.