Semi-Supervised Answer Extraction from Discussion Forums

Rose Catherine, Rashmi Gangadharaiah, Karthik Visweswariah, Dinesh Raghu
IBM Research
Bangalore, India
{rosecatherinek, rashgang, v-karthik, diraghu1} @in.ibm.com

Abstract

Mining online discussions to extract answers is an important research problem. Methods proposed in the past used supervised classifiers trained on labeled data. But, collecting training data for each target forum is labor intensive and time consuming, thus limiting their deployment. A recent approach had proposed to extract answers in an unsupervised manner, by taking cues from their repetitions. This assumption however, does not hold true in many cases. In this paper, we propose two semi-supervised methods for extracting answers from discussions, which utilize the large amount of unlabeled data available, alongside a very small training set to obtain improved accuracies. We show that it is possible to boost the performance by introducing a related, but parallel task of identifying acknowledgments to the answers. The accuracy achieved by our approaches surpass the baselines by a wide margin, as shown by our experiments.

1 Introduction

Online discussion forums, also known as community question answering (CQA) sites, are internet sites that provide a medium for users to discuss and share information on a wide range of topics. Due to their vast popularity, gradually, they have aggregated a massive collection of discussion data. Mining such forums have numerous applications such as improving question–answer (QA) retrieval (Cong et al., 2008), learning important insights like features of products that are drawing negative reviews (Lakkaraju et al., 2011) or discovering longstanding unresolved severe technical issues (Gangadharaiah and Catherine, 2012) etc. For this reason, substantial research effort has been directed at mining discussions, in recent times. In this paper, we focus on the specific problem of extracting answers from these discussions.

In forums, typically a user starts a discussion by posting a question to which multiple members of the forum suggest answers. The discussion evolves into a complex multi-party conversation as the question gets refined, with additional details specified, clarifications sought, multiple answers provided, frequent digressions, and occasional follow-up discussions and acknowledgments, altogether spanning several pages. Answers easily get buried deep within this and locating them automatically is far from straightforward.

In this paper, we propose two semi-supervised approaches that require only a very small amount of training data (only 3 manually tagged discussion threads) and achieve high accuracy levels by using the available unlabeled data. With this, we eliminate the need to collect vast amounts of training data, thus aiding faster deployment for new domains. Specifically, our contributions are:

- *A semi-supervised answer extraction method for discussions:* This paper makes the first attempt at extracting answers from discussions in a semi-supervised manner. We show how existing features can be engineered into a co-training framework to accomplish this.
- *A parallel co-training method to leverage acknowledgments for improved answer extraction accuracy:* We motivate and demonstrate that it is possible to improve the performance tremendously by introducing a related task of identifying acknowledgments in the discussions, which we run as a parallel task alongside the main answer extraction task (Section 5).
- We demonstrate that with a very small training data and by using the available unlabeled data, it is possible to extract answers from forums with an accuracy that is substantially better than extracting them in an unsupervised manner or in a fully supervised setting.

International Joint Conference on Natural Language Processing, pages 1–9,
Nagoya, Japan, 14-18 October 2013.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 sets the terminology and introduces the co-training framework, which is used throughout this paper. Section 4 details how the co-training framework can be applied to the answer extraction task. Section 5 introduces the acknowledgment extraction task in a parallel co-training framework. Experiments and results are discussed in Section 6 followed by conclusions in Section 7.

2 Related work

Research in the area of extracting question and answers from online forums, has grown considerably. Almost all approaches proposed so far for this task are supervised learning methods. Ding et al. (2008), Kim et al. (2010), Raghavan et al. (2010) and Kim et al. (2012) employed Conditional Random Fields, Hong and Davison (2009), Huang et al. (2007) and Catherine et al. (2012) used Support Vector Machines (SVM), Shrestha and McKeown (2004) learnt rules using Ripper, and Yang et al. (2009) used Struct SVMs for extracting answers. The obvious downside to these methods is that for any new domain or forum, substantial amounts of manually labeled training examples have to be collected. This is usually time consuming and costly. Gandhe et al. (2012) proposed an approach for adapting an answer extractor trained on one domain to another, by separating out the lexical characteristics of an answer from its domain relevance. However, learning the lexical characteristics still required a training set.

A recent work by Cong et al. (2008) proposed an unsupervised method using PageRank-style random walks on a graph representation of the discussion, with the hypothesis that inter-candidate similarities can improve accuracy of the answer extraction task. The intuition is that posts that bear more resemblance to other posts in the thread have higher chances of being answers. However, in a lot of discussion forums, especially those related to troubleshooting and problem resolution, we found that this assumption usually does not hold. An answer that was suggested earlier in the discussion is not usually suggested again – only new ones or a modification of the same would appear. A general observation here was that posts that had similar content as other posts were found to be others complaining about the same issue. This was also noted by Gandhe et al. (2012) and Catherine et al. (2012). Nevertheless, (Cong et al., 2008) is the only work so far, that sought to extract answers without supervision.

One of the methods proposed in this paper that uses a parallel acknowledgment classification task, belongs to the family of Multi-Task Learning (MTL) (Caruana, 1997) since what is learned for each task is used to improve the other task. However, to the best of our knowledge, this is the first work that proposes a MTL-type answer classifier for forums in a semi-supervised setting. Cross-Training (Sarawagi et al., 2003) is a related methodology which improves classification performance on one taxonomy by accessing labels from another taxonomy for the *same* document. Our method differs because, the answer and acknowledgment labels are on *different* posts.

Some other closely related works are listed below; however, their focus is different from the task proposed in this paper. Jijkoun and de Rijke (2005) proposed a method to automatically extract question–answer pairs from FAQ pages using formatting cues. Since it is known that the entry following the question is definitely an answer, they did not have to classify the entries. Sarencheh et al. (2010) proposed a semi-automatic wrapper induction method for extracting different structural components of a discussion, like the time of posting, author name, content of the post etc. Answer retrieval is another closely related task where the emphasis is on retrieving the most relevant post (Xue et al., 2008). The scope of our paper, however, is limited to tagging posts in forum discussions as answers or not.

3 Preliminaries

3.1 Terminology and Scope

A discussion in an online forum is created when a user posts a question. Other members of the forum reply to this post or to other replies, thereby evolving the discussion. A sample discussion with 7 posts including 2 answers and 2 acknowledgments, is shown in Figure 1. In this paper, we use the terms `discussion` and `thread` (as in, a thread of discussion) interchangeably.

An answer is typically spread over multiple sentences within the same post, especially in the case of non-factoid answers. It would have been ideal, if the system extracted answers at the granularity of a sentence. However, the inter-annotator agreement[1] for answer sentences in our dataset (Section 6) was a mere 0.19. Hence, we extract answers at

[1] http://en.wikipedia.org/wiki/Cohen's_kappa

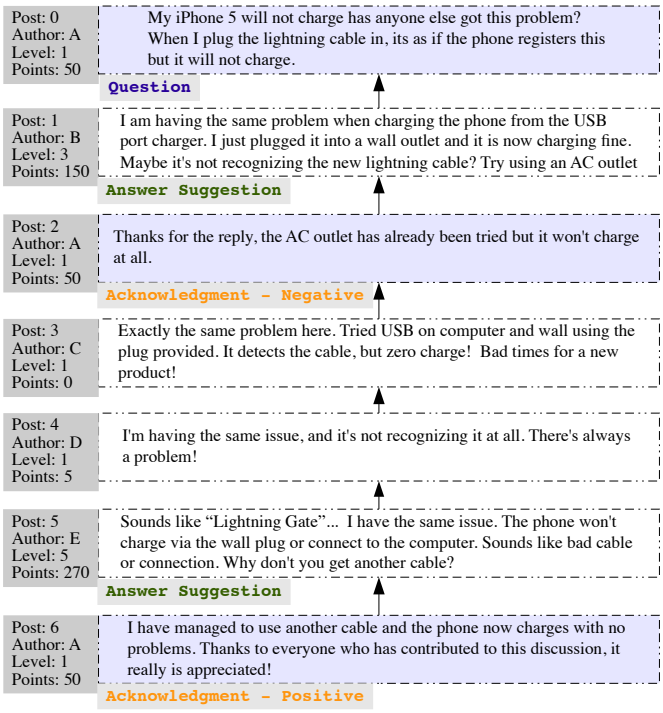

Figure 1: A Sample Discussion

a post level – a post is classified as an answer if any sentence within it suggests a solution.

Digressions are very common in such community question answering systems. We do not attempt to find these questions or separate out the sub-discussions. Question detection as well as disentangling multi-party discussions, is a well researched area (Cong et al., 2008; Elsner and Charniak, 2010), and is outside the scope of this paper. For the purposes of this paper, the first post is the question and we attempt to find answers to only this question. Answers to other questions within the discussion are negative examples.

3.2 Co-Training Methodology

Co-Training introduced by Blum and Mitchell (1998), is a general framework for semi-supervised classification, where the features for classifying each data point can be partitioned into two distinct sets or views. The views are such that either of them is sufficient to classify any datapoint, had there been enough training examples.

The algorithm proceeds in two half-steps: in iteration i, the current set of labeled examples L^i (initially, a very small set) is used to train a classifier C_1 that uses only one view v_1 of each training instance and another classifier C_2 that uses only view v_2. C_1 and C_2 are then used to classify the unlabeled points, and the most confident m predictions are moved from the unlabeled pool U to the set of labeled examples, which are used

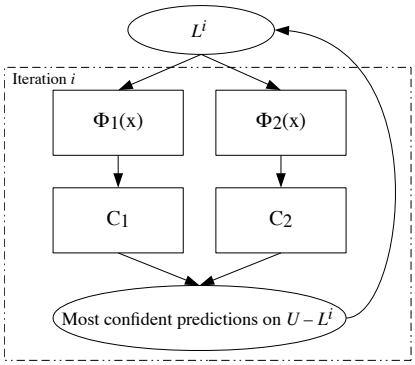

Figure 2: Co-Training Framework

for training in the $i + 1^{th}$ iteration. Essentially, each classifier teaches the other by providing examples which the other would have misclassified. Figure 2 shows this workflow, where Φ_1 and Φ_2 are the feature vectors of the input corresponding to the two views v_1 and v_2. The paper showed that when the two views are independent given the label of the data point (conditional independence), any initial weak predictor can be boosted to a high accuracy using unlabeled examples by co-training. This was empirically evaluated for a webpage classification task where v_1 was the set of words in the webpage and v_2, the anchor texts of all links pointing to that page. Co-training framework is widely used in many text mining tasks like parsing (Sarkar, 2001), machine translation (Callison-Burch, 2002) and for creating parallel corpora (Callison-Burch and Osborne, 2003).

4 Answer Extraction by Co-Training: ANS CT Model

To apply the co-training framework to the task of answer extraction, we need two independent views of the data. Prior work in supervised answer extraction from forum discussions have reported good accuracies when using features constructed from the structure of the thread (Ding et al., 2008; Hong and Davison, 2009; Kim et al., 2010; Catherine et al., 2012). This provides us with one of the views, which we refer to as the STRUCT view.

Cong et al. (2008) had previously used pattern mining for the related task of question sentence extraction. Similarly, Jindal and Liu (2006) had used pattern mining for identifying comparative sentences in a supervised learning setting. We mine patterns on the sentences of the posts and employ it as the second view, which we refer to as the PATTERN view. The exact set of features are explained in the subsections below.

Note that we have used only a modest set of features in both STRUCT and PATTERN views, to highlight the effect of co-training in improving the answer extraction accuracy.

4.1 STRUCT view

Compared to a general text document, discussion threads have a structure, which can be used to construct features for classification. The features that we use, referred to as STRUCT features henceforth, are listed in Table 1. All the features are eventually converted to binary attributes for the experiments, where numerical attributes are grouped into a suitable number of buckets. Each binary value corresponds to a dimension in the STRUCT view, Φ^i_{struct}, which is 1 if that attribute-value was present in the post; else, is set to 0.

STRUCT **Feature**	**Description**
`Author Rating`	A forum specific value measuring the expertise of the author. Could be numerical (e.g. 50 points) or categorical (e.g. `Expert`).
`Relative Post Position`	The position of the post with respect to the thread. It is grouped into `Beginning`, `Middle` and `End`.
`Post Rating`	A measure of how informative the post is. Could be numerical (e.g. 50 votes) or categorical (e.g. `Helpful`).

Table 1: STRUCT features for a post

4.2 PATTERN view

Consider the below snippets from different discussion threads. Some words have been intentionally masked to illustrate that it is possible to identify to a considerable extent, that these are answer suggestions from the structure of the sentence and without regard to the context or the question.

... You can see if X will solve it ...
... Try resetting your X with the Y turned off and then turn it back on after the X is fully booted back up ...
... Go to A -> B -> C and toggle the D mode ...
... X is no longer supported by Y ...

The PATTERN view uses a pattern mining module, which mines the answer posts in L^i to discover the most frequent sequential patterns, FP^i for iteration i. Each such discovered pattern $p \in FP^i$ corresponds to a dimension in the PATTERN view, $\Phi^i_{pattern}$, which is 1 if p matches (is subsequence of) any sentence of the post.

We implemented the PrefixSpan (Pei et al., 2001) algorithm for mining sequential patterns, but with the following modifications to contain the blow up in the number of patterns:

- Variable Minimum Support: the number of items in which a pattern appears is called its *support*, and minimum support, `min_sup` is an input parameter that determines whether a pattern is frequent enough. We set `min_sup` to $max(\texttt{min\_sup}_0, \texttt{frac} \times numItems^i)$, where `frac` is a preset fraction, set to 0.03 in our experiments, $numItems^i$ is the number of items being mined in iteration i, and $\texttt{min\_sup}_0$ is a default minimum, set to 3 in our experiments.
- Pattern Length: only patterns of length at least `min_len`, set to 3 in our experiments, are acceptable.
- Item Gap: the items of a frequent pattern are sequential, but not consecutive, thus allowing PrefixSpan to pick items that are arbitrary number of items apart (gap). We constrain the gap between items of a pattern to a maximum of `max_gap`, set to again 3 in our experiments.

Posts are mined at a sentence level, for which we use OpenNLP[2] sentence detector.

4.2.1 Text Pre-Processing

We found that using the exact words limited the number of frequent patterns that could be found. To minimize this problem, we used Part-Of-Speech (POS) tags of the words to:

- Replace all nouns with their POS tags.
- Replace all verbs with its root/stemmed (using Porter stemmer (Porter, 1980)) form and its POS tag. For example, `restarting` becomes `restart VBG`. We let PrefixSpan pick the verb-stem and/or the POS tag according to their support.

All words are lowercased. A discussion on the set of patterns that were detected is in Section 6.3.

5 Leveraging Acknowledgment Signals: ANS-ACK PCT Model

In this section, we motivate and introduce a related task of extracting acknowledgments in forum discussions and inducing signals from them to improve the accuracy of the answer extraction task.

Merriam-Webster[3] defines an acknowledgment as a recognition or favorable notice of an act or achievement. Acknowledgment is an inevitable component of any conversation, especially, when it evolves around seeking assistance. And so they find their place in forum discussions too. Consider the below snippets taken from replies by question

[2] http://opennlp.apache.org
[3] http://www.merriam-webster.com

authors. They are grouped according to their polarity – Positive, Negative and Neutral.

Positive: author reports that the suggestion solved the issue.

... Great! That solved it! Thanks a bunch ...
... Thanks for your help. Finally got it working ...
... Switching on X did the trick. Now I can Y without any problem ...
... Thanks a lot guys. X solved my woes. I must have Y-ed it by mistake at some point ...

Negative: the suggestion did not solve the issue.

... That didn't help. Any other suggestions? ...
... I tried that. It is still showing X ...
... Getting the same X. Thanks anyway ...
... Thanks for your advice. Unfortunately, it didn't help!

Neutral: it is not clear if the issue was solved, but the statement is an acknowledgment nevertheless

... Thanks for the reply ...
... I will try that ...
... Thanks for the helpful advice. Hope resetting X properly will fix my problem ...
... I'm reinstalling X. Will keep you posted ...

Similar to the case of answer sentences in Section 4, the above examples can be easily identified as acknowledgments and it is fairly clear that the posts to which the above sentences are replies, are answer suggestions. Note that this can be determined without knowing the contents of the latter, if we can assume that the reply-to relation of the posts is known. This however is not always the case. In a small study conducted, we found that only 75% of forums displayed or had the required information in the html of the webpage for constructing the reply-to relation of the posts, out of 12 technical forums that we inspected. In the absence of this information, (Wang et al., 2011; Seo et al., 2009; Wang and Rosé, 2010) propose techniques to automatically recover the structure. For the purposes of this paper, we assume that the reply-to structure of the discussion thread is given.

ANS-ACK PCT (ANSwer ACKnowledgment Parallel Co-Training) aims to leverage signals from acknowledgment posts, to better identify answer posts. We cast this as another instance of semi-supervised learning task (another co-training instance) which runs in parallel to the main answer extraction instance of co-training. Hence the name, PARALLEL co-training.

It is worth listing down some of the design decisions for this choice of approach:

(i) There is no public dataset available to train an acknowledgment classifier. So, it is important to note here that the task of detecting acknowledgments cannot be fully supervised where a large amount of training data is collected for the specific domain; this will defeat the entire purpose of semi-supervised answer extraction.

(ii) For the initial small training set required for the semi-supervised approach, we do not label additional threads. Instead, we create a training set from the initial training set of the answer extraction task by marking replies from the question author to posts that are answers, as positive examples. Other replies from the question author become negative examples. To avoid getting influenced by digressions, we do not consider replies from other authors.

(iii) The reader might suggest using acknowledgment as one of the views within the co-training instance of answer detection, instead of two parallel co-training instances. i.e. to mark all posts that have an acknowledgment as an answer in that view. Here, we would like to point out that acknowledgment is a strong indicator only when it is available. In other words, even if we learn to classify acknowledgment posts perfectly, it cannot classify all answer posts perfectly since not all answers are acknowledged. In our test set (Section 6), there were 559 answers, but only 173 of them had any reply from the question author (30.9%), of which only 72% were actually acknowledgments, as found through manual inspection (the rest had to do with refining the question, requesting clarification on the answer, etc.). So, the hope is to learn how to use the signal when it is available, and not rely on it exclusively by using it as one of the two views of answer co-training.

The acknowledgment extraction uses the same two views – STRUCT and PATTERN – for its co-training instance, similar to the ANS CT model of Section 4, to generate the views, ${}^{ack}\Psi^i_{struct}$ and ${}^{ack}\Psi^i_{pattern}$, respectively. Except that here, positive examples are the posts that are acknowledgments, as obtained by Point (ii) above.

5.1 Parallel Co-Training for Answer Extraction

Parallel Co-Training is a method for semi-supervised learning where there are two (or more) co-training instances corresponding to different, but related learning tasks running side by side, where in iteration i, each task can induce features based on the current state of the system. i.e. using the outcome of iteration $i-1$ of other tasks. Figure 3 depicts Parallel Co-Training for the specific case of answer extraction, where:

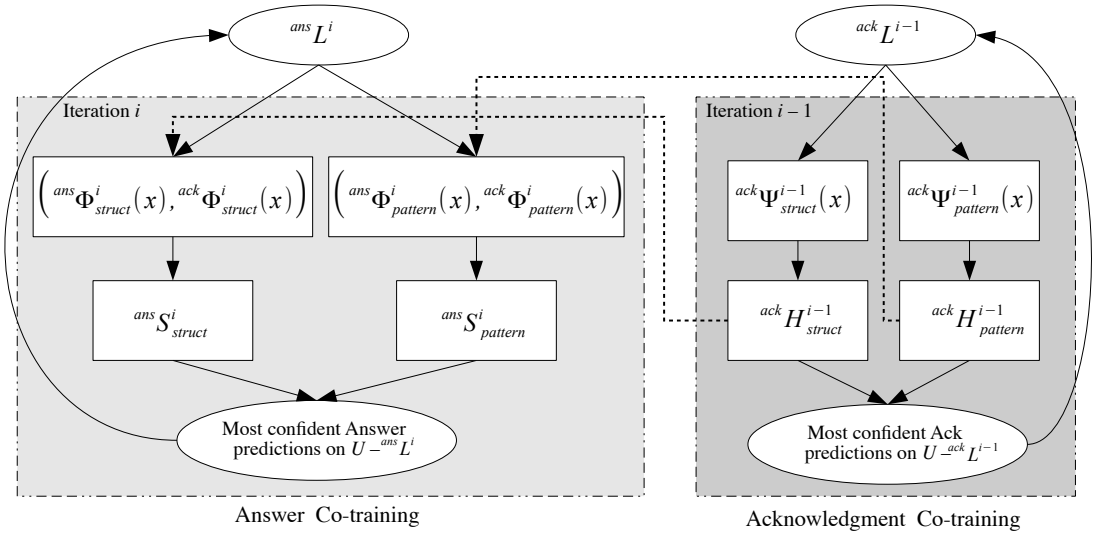

Figure 3: Parallel Co-Training Methodology

- Two tasks run in parallel:
  1. Answer Co-training: the main task which learns to classify each post as `ans` or $\overline{\texttt{ans}}$.
  2. Acknowledgment Co-training: the auxiliary task which learns to classify each post as `ack` or $\overline{\texttt{ack}}$.
- In iteration i, Answer Co-training uses the Acknowledgment classifiers, ${}^{ack}H_{:}^{i-1}$ of the $i-1^{th}$ iteration, to induce more features (detailed in Section 5.2), ${}^{ack}\Phi_{:}^{i}$ which is then concatenated to its original feature vector, ${}^{ans}\Phi_{:}^{i}$, to get the new feature vector $\left({}^{ans}\Phi_{:}^{i}, {}^{ack}\Phi_{:}^{i}\right)$ (we use $\left(\vec{A}, \vec{B}\right)$ to represent concatenation of two vectors, where the length of the new vector is $\left|\vec{A}\right| + \left|\vec{B}\right|$). The concatenated feature vector is then used to train the answer classifiers ${}^{ans}S_{:}^{i}$ of the i^{th} iteration.
- The STRUCT view of Answer Co-training uses predictions from the acknowledgment classifier that was trained on the STRUCT view of Acknowledgment Co-training, and similarly for the PATTERN view, so that the concatenated features vectors still remain independent.

5.2 Inducing Features from Acknowledgments

Given a thread and acknowledgment tags on some of the posts, the most obvious feature that can be induced on an answer post is a `hasAck` feature which is `True` if any child of this post is marked as an acknowledgment; else `False`. All features that we generated are listed in Table 2.

ACK Feature	Description
`Has Ack`	`True` if this post has a reply that is tagged as an acknowledgment; else `False`.
`Ack Distance`	The number of posts, in the chronological order, between this post and its acknowledgment.
`Last Ack Distance`	The number of posts, in the chronological order, between this post and the last acknowledgment post in the thread.

Table 2: Features induced from Acknowledgments

6 Experiments

We crawled about 140K threads from Apple Discussions[4]. From these, after discarding those with no replies, 303 threads were randomly chosen, and manually tagged. The inter-annotator agreement[5] between 3 annotators for this task was 0.71. For the experiments, the training set had 3 of the tagged threads and the remaining 300 formed the test set, the statistics of which are in Table 3.

Statistics	Training Set	Test Set
No. of Threads	3	300
Avg. Length of Threads	6.3	5.8
Avg. Answers per Thread	1.9	1.8
Fraction of Answers with Question Author's reply[6]	47.4%	30.9%

Table 3: Statistics of the Training and Test Sets

We used Support Vector Machines (Vapnik, 1995) (implementation from the LibSVM[7] library) for all the individual classifiers, ${}^{ans}S_{:}^{i}$ and

[4] https://discussions.apple.com
[5] http://en.wikipedia.org/wiki/Cohen's_kappa
[7] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

$^{ack}H^i_\cdot$, used in the different views of the co-training instances of ANS CT and ANS-ACK PCT models.

6.1 Study of Improvement in Answer Extraction Accuracy

	STRUCT	PATTERN	COMBINED
SVM	1.1%	2.5%	28.6%
ANS CT	55.6%	55.6%	55.6%
ANS-ACK PCT	63.7%	63.6%	67.8%

Table 4: F Scores for the Answer Extraction task

To demonstrate the benefits of co-training, we first trained a supervised classifier (SVM) on the training set for answer extraction, separately on the two views – STRUCT and PATTERN. With such a little amount of training data, the classifiers gave unimpressive F scores (van Rijsbergen, 1979), shown in the first row of Table 4. The COMBINED classifier is a combination of the individual STRUCT and PATTERN classifiers, computed as: $(P(\texttt{ans}|S_{combined}) \propto P(\texttt{ans}|S_{struct}) \times P(\texttt{ans}|S_{pattern}))$; and similarly for $\overline{\texttt{ans}}$. The post is tagged as ans if $P(\texttt{ans}|S_{combined}) \geq P(\overline{\texttt{ans}}|S_{combined})$. Else, it is $\overline{\texttt{ans}}$.

Next, we performed 40 iterations of co-training and in each step, 5 threads with the most confident predictions were added by each view from the unlabeled pool to the training set. If more than one thread had the same confidence, any one thread was chosen randomly. The accuracies achieved by ANS CT after the final iteration (averaged over 3 runs) is listed in Table 4. Clearly, both STRUCT and PATTERN classifiers drastically improved their F scores and the COMBINED classifier showed a substantial 94% improvement over the SVM baseline. The growth of F score of the two sub-classifiers as the co-training proceeds, is plotted in Figure 4. It can be seen that both the classifiers reached their best within 10 iterations and did not improve any further.

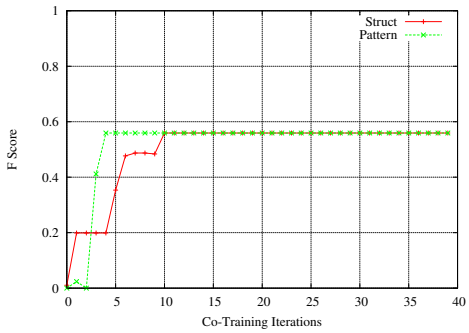

Figure 4: ANS CT: F-scores of the STRUCT and PATTERN sub-classifiers after each iteration

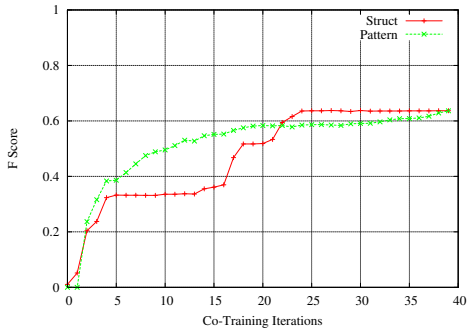

Figure 5: ANS-ACK PCT: F scores of the STRUCT and PATTERN sub-classifiers of the *Answer Classifier* after each iteration

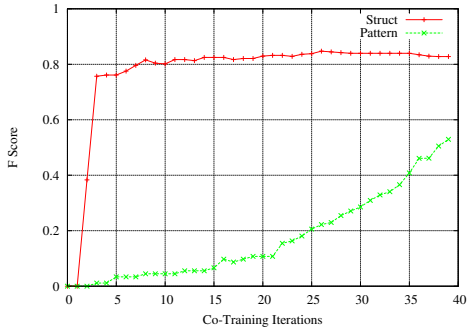

Figure 6: ANS-ACK PCT: F scores of theSTRUCT and PATTERN sub-classifiers of the *Acknowledgment Classifier* after each iteration

For ANS-ACK PCT, we performed 40 co-training iterations similar to ANS CT; the number of threads chosen after each iteration was similarly set to 5, for both answer and acknowledgment instances. The answer classification accuracies achieved by ANS-ACK PCT after the final iteration (averaged over 3 runs) is also listed in Table 4. Similar to ANS CT, both STRUCT and PATTERN classifiers improved their F scores significantly. The COMBINED classifier showed a substantial 137% improvement over the SVM baseline and 22% improvement over ANS CT. The F score growth of the answer and the acknowledgment classifiers as the co-training proceeds, is plotted in Figure 5 and Figure 6 respectively. Unlike Figure 4, the answer classifiers in Figure 5 continue to improve even after 10 iterations, since the acknowledgment instance is supplying it with more signals. In Figure 6, the PATTERN sub-classifier of the acknowledgment classifier constantly improved throughout the 40 iterations even though its STRUCT counterpart stabilized in about 10 iterations. The F scores of the acknowledgment classifiers at the end of the final iteration was 82.8%, 52.9% and 81.9% respectively for STRUCT, PATTERN and COMBINED.

6.2 Comparison between approaches

	Precision	Recall	F score
SVM - STRUCT	75.0%	0.5%	1.1%
SVM - PATTERN	25.9%	1.3%	2.5%
ANS CT	40.6%	88.0%	55.6%
ANS-ACK PCT	56.8%	84.1%	67.8%
CONG	29.7%	55.6%	38.7%

Table 5: Comparison of accuracy measures of different methods for the answer classification task

Graph Propagation based Answer Extraction by Cong et al. (Cong et al., 2008) is an unsupervised method for extracting answers from discussion forums. It is based on the premise that a correct answer will be repeated often within the discussion and thus, the similarity of the post to other posts can be used as a measure of their "answer-ness". We used our own implementation of this algorithm, referred to as CONG in the experiments . The similarity between posts is computed using Kullback-Leibler divergence (Kullback, 1997), which was reported by the authors to have given the best performance. The Precision, Recall and F scores of CONG are compared with those of the proposed methods in Table 5.

Table 5 shows that both proposed methods perform substantially better than CONG: ANS CT exceeds it by 43.6% and ANS-ACK PCT surpasses it by 75.1% (F score). Consequently, we can conclude that with very small training data, it is possible to achieve high accuracies compared to extracting them in an unsupervised manner.

6.3 Discussion: Answer and Acknowledgment Patterns

This section studies the patterns that were mined from answers and acknowledgments, which reveal the types of sentence structures that frequently appear in them. Some of the interesting answer patterns are listed in Table 6. They are grouped into Imperative, Factual, Conditional and Questions, based on manual inspection. Similarly, the acknowledgment sentences also showed interesting patterns, manually grouped into Action and Others, listed in Table 7. From inspecting the answer and acknowledgment patterns, we conjecture that it should be possible to build classifiers based on rules defined over the structure of the sentence, without requiring access to a training set.

Pattern Type	Examples
Imperative Sentence	1. go - to - NNS - NN - on 2. you - can - VB - NN 3. VBG - your - NN - NN 4. VB - to - NNS - NN 5. VBG - your - NN 6. check - NN - NN
Fact	7. is - VBZ - not - NN
Conditional Statement	8. if - you - VBP - NN
Questions	9. have - VB - you - tri - VBN - VBG - NN 10. have - you - VBN - VBG - NN

Table 6: Mined Answer Patterns

Pattern Type	Examples
Action	1. i - VBP - to - VB 2. i - VBP - NN - it 3. i - not - VB - NN 4. i - have - VBP - NN, 5. i - am - VBP - VBG - NN
Others	6. but - i - VBP 7. i - am - VBP - sure

Table 7: Mined Acknowledgment Patterns

7 Conclusion and Future Work

In this paper, we proposed two semi-supervised methods for extracting answers from discussion threads. We showed how the structural features and sentence construction patterns could be engineered into a co-training setting such that by using a very small training set, and the large amount of unlabeled data available, answers could be extracted with high accuracy, substantially surpassing that attained by an unsupervised method. To demonstrate the benefits of our method, we also showed that completely supervised methods would fail to train a decent model with the very little training data that we used.

In one of our methods, we motivated and introduced a related task of identifying acknowledgments to the answers, which was cast in a parallel co-training setting. We proposed new features which the answer labeling instance could induce from the acknowledgment instance. Our experiments showed that having access to this view of the discussion thread substantially improved the answer extraction accuracy.

Our work opens up new directions of research. In the parallel co-training setting, other than inducing features, the co-training instances are essentially independent. In future, we plan to extend it such that the two instances would collaboratively label new threads; this should lead to higher gains since the instances would now strive to achieve higher coherence between their labels. Also, extending the method to extract answers at a lower granularity like a snippet or a sentence, instead of at a post level would be advantageous for domains that have more factoid type answers.

References

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proc. eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100.

C. Callison-Burch and M. Osborne. 2003. Bootstrapping parallel corpora. In *Proc. HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 44–49. Association for Computational Linguistics.

C. Callison-Burch. 2002. Co-training for statistical machine translation. In *Proc. of the 6th Annual CLUK Research Colloquium*.

R. Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75, July.

R. Catherine, A. Singh, R. Gangadharaiah, D. Raghu, and K. Visweswariah. 2012. Does similarity matter? the case of answer extraction from technical discussion forums. In *Proc. 24th International Conference on Computational Linguistics*, COLING, pages 175–184.

G. Cong, L. Wang, C. Lin, Y. Song, and Y. Sun. 2008. Finding question-answer pairs from online forums. In *The 31st Annual International ACM SIGIR Conference*, pages 467–474.

S. Ding, G. Cong, C. Y. Lin, and X. Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *Meeting of the Association for Computational Linguistics (ACL)*.

M. Elsner and E. Charniak. 2010. Disentangling chat. *Computational Linguistics*, 36:389–409.

A. Gandhe, D. Raghu, and R. Catherine. 2012. Domain adaptive answer extraction for discussion boards. In *Proc. 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 501–502. ACM.

R. Gangadharaiah and R. Catherine. 2012. Prism: discovering and prioritizing severe technical issues from product discussion forums. In *Proc. 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1627–1631.

L. Hong and B. D. Davison. 2009. A classification-based approach to question answering in discussion boards. In *Proc. 32nd Annual Intl ACM SIGIR Conf. on Research and Dev. in Information Retrieval*.

J. Huang, M. Zhou, and D. Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *Proc. 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 423–428.

V. Jijkoun and M. de Rijke. 2005. Retrieving answers from frequently asked questions pages on the web. In *Proc. 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 76–83.

Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *In Proc. 29th SIGIR*.

S. N. Kim, L. Wang, and T. Baldwin. 2010. Tagging and linking web forum posts. In *Proc. Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 192–202. Association for Computational Linguistics.

S. N. Kim, L. Cavedon, and T. Baldwin. 2012. Classifying dialogue acts in multi-party live chats. In *Proc. 26th Pacic Asia Conference on Language, Information and Computation*, pages 463–472.

S. Kullback. 1997. *Information Theory and Statistics*. Dover Publications.

H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu. 2011. Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In *Proc. 11th SIAM International Conference on Data Mining*, SDM '11, pages 498–509.

J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. 17th International Conference on Data Engineering*. IEEE Computer Society.

M.F. Porter. 1980. An algorithm for suffix stripping. In *Program*.

P. Raghavan, R. Catherine, S. Ikbal, N. Kambhatla, and D. Majumdar. 2010. Extracting problem and resolution information from online discussion forums. In *Proc. 16th International Conference on Management of Data*, COMAD.

S. Sarawagi, S. Chakrabarti, and S. Godbole. 2003. Cross-training: learning probabilistic mappings between topics. In *Proc. ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 177–186.

S. Sarencheh, V. Potdar, E. Yeganeh, and N. Firoozeh. 2010. Semi-automatic information extraction from discussion boards with applications for anti-spam technology. In *Computational Science and Its Applications - ICCSA 2010*, volume 6017 of *Lecture Notes in Computer Science*, pages 370–382. Springer Berlin Heidelberg.

A. Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proc. second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01.

J. Seo, B. Croft, and D. A. Smith. 2009. Online community search using thread structure. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1907–1910. ACM.

L. Shrestha and K. McKeown. 2004. Detection of question-answer pairs in email conversation. In *Proc. 20th International Conference on Computational LINGuistic(COLING)*.

C. J. van Rijsbergen. 1979. *Information Retrieval (2nd ed.)*. Butterworth.

V. N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

Y. Wang and C. P. Rosé. 2010. Making conversational structure explicit: identification of initiation-response pairs within online discussions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 673–676. Association for Computational Linguistics.

H. Wang, C. Wang, C. Zhai, and J. Han. 2011. Learning online discussion structures by conditional random fields. In *Proc. 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 435–444.

X. Xue, J. Jeon, and W. B. Croft. 2008. Retrieval models for question and answer archives. In *Proc. 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 475–482. ACM.

W. Yang, Y. Cao, and C. Lin. 2009. A structural support vector method for extracting contexts and answers of questions from online forums. In *Proc. 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 514–523.