Unsupervised Solution Post Identification from Discussion Forums

Deepak P
IBM Research - India
Bangalore, India
deepak.s.p@in.ibm.com

Karthik Visweswariah
IBM Research - India
Bangalore, India
v-karthik@in.ibm.com

Abstract

Discussion forums have evolved into a dependable source of knowledge to solve common problems. However, only a minority of the posts in discussion forums are solution posts. Identifying solution posts from discussion forums, hence, is an important research problem. In this paper, we present a technique for unsupervised solution post identification leveraging a so far unexplored textual feature, that of lexical correlations between problems and solutions. We use translation models and language models to exploit lexical correlations and solution post character respectively. Our technique is designed to not rely much on structural features such as post metadata since such features are often not uniformly available across forums. Our clustering-based iterative solution identification approach based on the EM-formulation performs favorably in an empirical evaluation, beating the only unsupervised solution identification technique from literature by a very large margin. We also show that our unsupervised technique is competitive against methods that require supervision, outperforming one such technique comfortably.

1 Introduction

Discussion forums have become a popular knowledge source for finding solutions to common problems. StackOverflow[1], a popular discussion forum for programmers is among the top-100 most visited sites globally[2]. Now, there are discussion forums for almost every major product ranging from automobiles[3] to gadgets such as those of Mac[4] or Samsung[5]. These typically start with a registered user posting a question/problem[6] to which other users respond. Typical response posts include solutions or clarification requests, whereas feedback posts form another major category of forum posts. As is the case with any community of humans, discussion forums have their share of inflammatory remarks too. Mining problem-solution pairs from discussion forums has attracted much attention from the scholarly community in the recent past. Since the first post most usually contains the problem description, identifying its solutions from among the other posts in the thread has been the focus of many recent efforts (e.g., (Gandhe et al., 2012; Hong and Davison, 2009)). Extracting problem-solution pairs from forums enables the usage of such knowledge in knowledge reuse frameworks such as case-based reasoning (Kolodner, 1992) that use problem-solution pairs as raw material. In this paper, we address the problem of unsupervised solution post identification[7] from discussion forums.

Among the first papers to address the solution identification problem was the unsupervised approach proposed by (Cong et al., 2008). It employs a graph propagation method that prioritizes posts that are (a) more similar to the problem post, (b) more similar to other posts, and (c) authored by a more authoritative user, to be labeled as solution posts. Though seen to be effective in identifying solutions from travel forums, the first two assumptions, (a) and (b), were seen to be not very

[1] http://www.stackoverflow.com

[2] http://www.alexa.com/siteinfo/stackoverflow.com

[3] http://www.cadillacforums.com/

[4] https://discussions.apple.com/

[5] http://www.galaxyforums.net/

[6] We use problem and question, as well as solution and answer interchangeably in this paper.

[7] This problem has been referred to as *answer extraction* by some papers earlier. However, we use *solution identification* to refer to the problem since *answer* and *extraction* have other connotations in the Question-Answering and Information Extraction communities respectively.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 155–164,
Baltimore, Maryland, USA, June 23-25 2014.

reliable in solution identification in other kinds of discussion boards. (Catherine et al., 2012) reports a study that illustrates that non-solution posts are, on an average, as similar to the problem as solution posts in technical forums. The second assumption (i.e., (b) above) was also not seen to be useful in discussion forums since posts that are highly similar to other posts were seen to be complaints, repetitive content being more pervasive among complaint posts than solutions (Catherine et al., 2013). Having exhausted the two obvious textual features for solution identification, subsequent approaches have largely used the presence of lexical cues signifying solution-like narrative (e.g., instructive narratives such as *"check the router for any connection issues"*) as the primary content-based feature for solution identification.

All solution identification approaches since (Cong et al., 2008) have used supervised methods that require training data in the form of labeled solution and non-solution posts. The techniques differ from one another mostly in the non-textual features that are employed in representing posts. A variety of high precision assumptions such as *solution post typically follows a problem post* (Qu and Liu, 2011), *solution posts are likely to be within the first few posts*, *solution posts are likely to have been acknowledged by the problem post author* (Catherine et al., 2012), *users with high authoritativeness are likely to author solutions* (Hong and Davison, 2009), and so on have been seen to be useful in solution identification. Being supervised methods, the above assumptions are implicitly factored in by including the appropriate feature (e.g., post position in thread) in the feature space so that the learner may learn the correlation (e.g., solution posts typically are among the first few posts) using the training data. Though such assumptions on structural features, if generic enough, may be built into unsupervised techniques to aid solution identification, the variation in availability of such features across forums limits the usage of models that rely heavily on structural features. For example, some forums employ chronological order based flattening of threads (Seo et al., 2009) making reply-to information unavailable; models that harness reply-to features would then have limited utility on identifying solutions within such flattened threads. On medical forums, privacy considerations may force forum data to be dumped without author information, making a host of author-id based features unavailable. On datasets that contain data from across forums, the model may have to be aware of the absence of certain features in subsets of the data, or be modeled using features that are available on all threads.

Our Contribution: We propose an unsupervised method for solution identification. The cornerstone of our technique is the usage of a hitherto unexplored textual feature, *lexical correlations between problems and solutions*, that is exploited along with language model based characterization of solution posts. We model the lexical correlation and solution post character using regularized translation models and unigram language models respectively. To keep our technique applicable across a large variety of forums with varying availability of non-textual features, we design it to be able to work with *minimal availability of non-textual features*. In particular, we show that by using post position as the only non-textual feature, we are able to achieve accuracies comparable to supervision-based approaches that use many structural features (Catherine et al., 2013).

2 Related Work

In this section, we provide a brief overview of previous work related to our problem. Though most of the answer/solution identification approaches proposed so far in literature are supervised methods that require a labeled training corpus, there are a few that require limited or no supervision. Table 1 provides an overview of some of the more recent solution identification techniques from literature, with a focus on some features that we wish to highlight. The common observation that most problem-solving discussion threads have a problem description in the first post has been explicitly factored into many techniques; knowing the problem/question is important for solution identification since author relations between problem and other posts provide valuable cues for solution identification. Most techniques use a variety of such features as noted in Section 1. SVMs have been the most popular method for supervised and semi-supervised learning for the task of solution identification.

Of particular interest to us are approaches that use limited or no supervision, since we focus on unsupervised solution identification in this paper.

Paper Reference	Supervision	Assumptions on Problem Position	Features other than Post Content Used	Learning Technique
(Qu and Liu, 2011)	Supervised	First Post likely to be problem	HMM assumes solution follows problem	Naive Bayes & HMM
(Ding et al., 2008)	Supervised	First Post	Post Position, Author, Context Posts	CRFs
(Kim et al., 2010)	Supervised	None	Post Position, Author, Previous Posts, Profile etc.	MaxEnt, SVM, CRF
(Hong and Davison, 2009)	Supervised	First Post	Post Position, Author, Author Authority	SVM
(Catherine et al., 2012)	Supervised	First Post	Post Position, Author, Problem Author's activities wrt Post	SVM
(Catherine et al., 2013)	Limited Supervision	First Post	Post Position/Rating, Author, Author Rating, Post Ack	SVMs & Co-Training
(Cong et al., 2008)	**Unsupervised**	None	Author, Author Authority, Relation to Problem Author	Graph Propagation
Our Method	**Unsupervised**	First Post	Post Position	Translation Models & LM

Table 1: Summary of Some Solution Identification Techniquess

The **only** unsupervised approach for the task, that from (Cong et al., 2008), uses a graph propagation method on a graph modeled using posts as vertices, and relies on the assumptions that posts that bear high similarity to the problem and other posts and those authored by authoritative users are more likely to be solution posts. Some of those assumptions, as mentioned in Section 1, were later found to be not generalizable to beyond travel forums. The semi-supervised approach presented in (Catherine et al., 2013) uses a few labeled threads to bootstrap SVM based learners which are then co-trained in an iterative fashion. In addition to various features explored in literature, they use acknowledgement modeling so that posts that have been acknowledged positively may be favored for being labeled as solutions.

We will use translation and language models in our method for solution identification. Usage of translation models for modeling the correlation between textual problems and solutions have been explored earlier starting from the answer retrieval work in (Xue et al., 2008) where new queries were conceptually expanded using the translation model to improve retrieval. Translation models were also seen to be useful in segmenting incident reports into the problem and solution parts (Deepak et al., 2012); we will use an adaptation of the generative model presented therein, for our solution extraction formulation. Entity-level translation models were recently shown to be useful in modeling correlations in QA archives (Singh, 2012).

3 Problem Definition

Let a thread $\mathcal{T}$ from a discussion forum be made up of t posts. Since we assume, much like many other earlier papers, that the first post is the problem post, the task is to identify which among the remaining $t-1$ posts are solutions. There could be multiple (most likely, different) solutions within the same thread. We may now model the thread $\mathcal{T}$ as $t-1$ post pairs, each pair having the problem post as the first element, and one of the $t-1$ remaining posts (i.e., reply posts in $\mathcal{T}$) as the second element. Let $\mathcal{C} = \{(p_1, r_1), (p_2, r_2), \ldots, (p_n, r_n)\}$ be the set of such problem-reply pairs from across threads in the discussion forum. We are interested in finding a subset $\mathcal{C}'$ of $\mathcal{C}$ such that most of the pairs in $\mathcal{C}'$ are problem-solution pairs, and most of those in $\mathcal{C} - \mathcal{C}'$ are not so. In short, we would like to find problem-solution pairs from $\mathcal{C}$ such that the F-measure[8] for solution identification is maximized.

4 Our Approach

4.1 The Correlation Assumption

Central to our approach is the assumption of lexical correlation between the problem and solution

[8] http://en.wikipedia.org/wiki/F1_score

texts. At the word level, this translates to assuming that there exist word pairs such that the presence of the first word in the problem part predicts the presence/absence of the second word in the solution part well. Though not yet harnessed for solution identification, the correlation assumption is not at all novel. Infact, the assumption that similar problems have similar solutions (of which the correlation assumption is an offshoot) forms the foundation of case-based reasoning systems (Kolodner, 1992), a kind of knowledge reuse systems that could be the natural consumers of problem-solution pairs mined from forums. The usage of translation models in QA retrieval (Xue et al., 2008; Singh, 2012) and segmentation (Deepak et al., 2012) were also motivated by the correlation assumption. We use an IBM Model 1 translation model (Brown et al., 1990) in our technique; simplistically, such a model m may be thought of as a 2-d associative array where the value $m[w_1][w_2]$ is directly related to the probability of w_1 occuring in the problem when w_2 occurs in the solution.

4.2 Generative model for Solution Posts

Consider a unigram language model $\mathcal{S}_S$ that models the lexical characteristics of solution posts, and a translation model $\mathcal{T}_S$ that models the lexical correlation between problems and solutions. Our generative model models the reply part of a (p, r) pair (in which r is a solution) as being generated from the statistical models in $\{\mathcal{S}_S, \mathcal{T}_S\}$ as follows.

- For each word w_s occuring in r,
  1. Choose $z \sim U(0, 1)$
  2. If $z \leq \lambda$, Choose $w \sim Mult(\mathcal{S}_S)$
  3. Else, Choose $w \sim Mult(\mathcal{T}_S^p)$

where $\mathcal{T}_S^p$ denotes the multionomial distribution obtained from $\mathcal{T}_S$ conditioned over the words in the post p; this is obtained by assigning each candidate solution word w a weight equal to $avg\{\mathcal{T}_S[w'][w]|w' \in p\}$, and normalizing such weights across all solution words. *In short, each solution word is assumed to be generated from the language model or the translation model (conditioned on the problem words) with a probability of λ and $1 - \lambda$ respectively, thus accounting for the correlation assumption.* The generative model above is similar to the proposal in (Deepak et al., 2012), adapted suitably for our scenario. We model non-solution posts similarly with the sole difference being that they would be sampled from the analogous models $\mathcal{S}_N$ and $\mathcal{T}_N$ that characterize behavior of non-solution posts.

Example: Consider the following illustrative example of a problem and solution post:

- *Problem:* I am unable to surf the web on the BT public wifi.
- *Solution:* Maybe, you should try disconnecting and rejoining the network.

Of the solution words above, generic words such as *try* and *should* could probably be explained by (i.e., sampled from) the solution language model, whereas *disconnect* and *rejoin* could be correlated well with *surf* and *wifi* and hence are more likely to be supported better by the translation model.

4.3 Clustering-based Approach

We propose a clustering based approach so as to cluster each of the (p, r) pairs into either the *solution* cluster or the *non-solution* cluster. The objective function that we seek to maximize is the following:

$$\sum_{(p,r)\in\mathcal{C}} \begin{cases} F((p,r),\mathcal{S}_S,\mathcal{T}_S) & \textit{if label((p,r))=S} \\ F((p,r),\mathcal{S}_N,\mathcal{T}_N) & \textit{if label((p,r))=N} \end{cases} \quad (1)$$

$F((p, r), \mathcal{S}, \mathcal{T})$ indicates the conformance of the (p, r) pair (details in Section 4.3.1) with the generative model that uses the $\mathcal{S}$ and $\mathcal{T}$ models as the language and translation models respectively. The clustering based approach labels each (p, r) pair as either solution (i.e., S) or non-solution (i.e., N). Since we do not know the models or the labelings to start with, we use an iterative approach modeled on the EM meta-algorithm (Dempster et al., 1977) involving iterations, each comprising of an E-step followed by the M-step. For simplicity and brevity, instead of deriving the EM formulation, we illustrate our approach by making an analogy with the popular K-Means clustering (MacQueen, 1967) algorithm that also uses the EM formulation and crisp assignments of data points like we do. K-Means is a clustering algorithm that clusters objects represented as multi-dimensional points into k clusters where each cluster is represented by the centroid of all its members. Each iteration in K-Means starts off with assigning each

	In K-Means	In Our Approach
Data	Multi-dimensional Points	(p, r) pairs
Cluster Model	Respective Centroid Vector	Respective $\mathcal{S}$ and $\mathcal{T}$ Models for each cluster
Initialization	Random Choice of Centroids	Models learnt using (p, r) pairs labeled using the Post Position of r
E-Step	$label(d) = \arg\min_i dist(d, centroid_i)$	$label((p, r)) = \arg\max_i F((p, r), \mathcal{S}_i, \mathcal{T}_i)$ (Sec 4.3.1), and learn solution word source probabilities (Sec 4.3.2)
M-Step	$centroid_i = avg\{d \mid label(d) = i\}$	Re-learn $\mathcal{S}_S$ and $\mathcal{T}_S$ using pairs labeled S $\mathcal{S}_N$ and $\mathcal{T}_N$ using pairs labeled N (Sec 4.3.3)
Output	The clustering of points	(p, r) pairs labeled as S

Table 2: Illustrating Our Approach wrt K-Means Clustering

data object to its nearest centroid, followed by re-computing the centroid vector based on the assignments made. The analogy with K-Means is illustrated in Table 2.

Though the analogy in Table 2 serves to provide a high-level picture of our approach, the details require further exposition. In short, our approach is a 2-way clustering algorithm that uses two pairs of models, $[\mathcal{S}_S, \mathcal{T}_S]$ and $[\mathcal{S}_N, \mathcal{T}_N]$, to model solution pairs and non-solution pairs respectively. At each iteration, the post-pairs are labeled as either solution (S) or non-solution (N) based on which pair of models they better conform to. Within the same iteration, the four models are then re-learnt using the labels and other side information. At the end of the iterations, the pairs labeled S are output as solution pairs. We describe the various details in separate subsections herein.

4.3.1 E-Step: Estimating Labels

As outlined in Table 2, each (p, r) pair would be assigned to one of the classes, solution or non-solution, based on whether it conforms better with the solution models (i.e., $\mathcal{S}_S$ & $\mathcal{T}_S$) or non-solution models ($\mathcal{S}_N$ & $\mathcal{T}_N$), as determined using the $F((p, r), \mathcal{S}, \mathcal{T})$ function, i.e.,

$$label((p, r)) = \arg\max_{i \in \{S,N\}} F((p, r), \mathcal{S}_i, \mathcal{T}_i)$$

$F(.)$ falls out of the generative model:

$$F((p, r), \mathcal{S}, \mathcal{T}) = \prod_{w \in r} \lambda \times \mathcal{S}[w] + (1 - \lambda) \times \mathcal{T}^p[w]$$

where $\mathcal{S}[w]$ denotes the probability of w from $\mathcal{S}$ and $\mathcal{T}^p[w]$ denotes the probability of w from the multinomial distribution derived from $\mathcal{T}$ conditioned over the words in p, as in Section 4.2.

4.3.2 E-Step: Estimating Reply Word Source

Since the language and translation models operate at the word level, the objective function entails that we let the models learn based on their fractional contribution of the words from the language and translation models. Thus, we estimate the proportional contribution of each word from the language and translation models too, in the E-step. The fractional contributions of the word $w \in r$ in the (p, r) pair labeled as solution (i.e., S) is as follows:

$$f^{(p,r)}_{\mathcal{S}_S}(w) = \frac{\mathcal{S}_S[w]}{\mathcal{S}_S[w] + \mathcal{T}^p_S[w]}$$

$$f^{(p,r)}_{\mathcal{T}_S}(w) = \frac{\mathcal{T}^p_S[w]}{\mathcal{S}_S[w] + \mathcal{T}^p_S[w]}$$

The fractional contributions are just the actual supports for the word w, normalized by the total contribution for the word from across the two models. Similar estimates, $f^{(p,r)}_{\mathcal{S}_N}(.)$ and $f^{(p,r)}_{\mathcal{S}_N}(.)$ are made for reply words from pairs labeled N. In our example from Section 4.2, words such as *rejoin* are likely to get higher $f^{(p,r)}_{\mathcal{T}_S}(.)$ scores due to being better correlated with problem words and consequently better supported by the translation model; those such as *try* may get higher $f^{(p,r)}_{\mathcal{S}_S}(.)$ scores.

4.3.3 M-Step: Learning Models

We use the labels and reply-word source estimates from the E-step to re-learn the language and translation models in this step. As may be obvious from the ensuing discussion, those pairs labeled as solution pairs are used to learn the $\mathcal{S}_S$ and $\mathcal{T}_S$ models and those labeled as non-solution pairs are

used to learn the models with subscript N. We let each reply word contribute as much to the respective language and translation models according to the estimates in Section 4.3.2. In our example, if the word *disconnect* is assigned a source probability of 0.9 and 0.1 for the translation and language models respectively, the virtual document-pair from (p, r) that goes into the training of the respective $\mathcal{T}$ model would assume that *disconnect* occurs in r with a frequency of 0.9; similarly, the respective $\mathcal{S}$ would account for *disconnect* with a frequency of 0.1. Though fractional word frequencies are not possible in real documents, statistical models can accomodate such fractional frequencies in a straightforward manner. The language models are learnt only over the r parts of the (p, r) pairs since they are meant to characterize reply behavior; on the other hand, translation models learn over both p and r parts to model correlation.

Regularizing the $\mathcal{T}$ models: In our formulation, the language and translation models may be seen as competing for "ownership" of reply words. Consider the post and reply vocabularies to be of sizes A and B respectively; then, the translation model would have $A \times B$ variables, whereas the unigram language model has only B variables. This gives the translation model an implicit edge due to having more parameters to tune to the data, putting the language models at a disadvantage. To level off the playing field, we use a regularization[9] operation in the learning of the translation models. The IBM Model 1 learning process uses an internal EM approach where the E-step estimates the alignment vector for each problem word; this vector indicates the distribution of alignments of the problem word across the solution words. In our example, an example alignment vector for *wifi* could be: $\{rejoin : 0.4, network : 0.4, disconnect : 0.1, \ldots\}$. Our regularization method uses a parameter τ to discard the long tail in the alignment vector by resetting entries having a value $\leq \tau$ to 0.0 followed by re-normalizing the alignment vector to add up to 1.0. Such pruning is performed at each iteration in the learning of the translation model, so that the following M-steps learn the probability matrix according to such modified alignment vectors.

The semantics of the τ parameter may be intuitively outlined. If we would like to allow alignment vectors to allow a problem word to align with upto two reply words, we would need to set τ to a value close to $0.5 (= \frac{1}{2})$; ideally though, to allow for the mass consumed by an almost inevitable long tail of very low values in the alignment vector, we would need to set it to slightly lower than 0.5, say 0.4.

[9]We use the word *regularization* in a generic sense to mean adapting models to avoid overfitting; in particular, it may be noted that we are not using popular regularization methods such as L1-regularization.

Alg. 1 ***Clustering-based Solution Identification***

Input. $\mathcal{C}$, a set of (p, r) pairs
Output. $\mathcal{C}'$, the set of identified solution pairs

Initialization
1. $\forall (p, r) \in \mathcal{C}$
2. $\quad if(r.postpos = 2)\ label((p, r)) = S$
3. $\quad else\ label((p, r)) = N$
4. Learn $\mathcal{S}_S$ & $\mathcal{T}_S$ using pairs labeled S
5. Learn $\mathcal{S}_N$ & $\mathcal{T}_N$ using pairs labeled N

EM Iterations
6. $while(not\ converged \wedge \#Iterations < 10)$
   E-Step:
7. $\quad \forall (p, r) \in \mathcal{C}$
8. $\quad label((p, r)) = \arg\max_i F((p, r), \mathcal{S}_i, \mathcal{T}_i)$
9. $\quad \forall w \in r$
10. $\quad\quad Estimate\ f^{(p,r)}_{\mathcal{S}_{label(p,r)}}(w)\ ,\ f^{(p,r)}_{\mathcal{T}_{label(p,r)}}(w)$
    M-Step:
11. $\quad$ Learn $\mathcal{S}_S$ & $\mathcal{T}_S$ from pairs labeled S using the $f^{(p,r)}_{\mathcal{S}_S}(.)\ f^{(p,r)}_{\mathcal{T}_S}(.)$ estimates
12. $\quad$ Learn $\mathcal{S}_N$ & $\mathcal{T}_N$ from pairs labeled N using the $f^{(p,r)}_{\mathcal{S}_N}(.)\ f^{(p,r)}_{\mathcal{T}_N}(.)$ estimates

Output
13. Output (p, r) pairs from $\mathcal{C}$ with $label((p, r)) = S$ as $\mathcal{C}'$

4.3.4 Initialization

K-Means clustering mostly initializes centroid vectors randomly; however, it is non-trivial to initialize the complex translation and language models randomly. Moreover, an initialization such that the $\mathcal{S}_S$ and $\mathcal{T}_S$ models favor the solution pairs more than the non-solution pairs is critical so that they may progressively lean towards modeling solution behaviour better across iterations. Towards this, we make use of a structural feature; in particular, adapting the hypothesis that solutions occur in the first N posts (Ref. (Catherine et al., 2012)), *we label the pairs that have the the reply from the second post (note that the first post is assumed to be the problem post) in the thread as a solution*

post, and all others as non-solution posts. Such an initialization along with uniform reply word source probabilities is used to learn the initial estimates of the $\mathcal{S}_S$, $\mathcal{T}_S$, $\mathcal{S}_N$ and $\mathcal{T}_N$ models to be used in the E-step for the first iteration. We will show that we are able to effectively perform solution identification using our approach by exploiting just one structural feature, the post position, as above. However, we will also show that we can exploit other features as and when available, to deliver higher accuracy clusterings.

4.3.5 Method Summary

The overall method comprising the steps that have been described is presented in Algorithm 1. The initialization using the post position (Ref. Sec 4.3.4) is illustrated in Lines 1-5, whereas the EM-iterations form Steps 6 through 12. Of these, the E-step incorporates labeling (Line 8) as described in Sec 4.3.1 and reply-word source estimation (Line 10) detailed in Sec 4.3.2. The models are then re-learnt in the M-Step (Lines 11-12) as outlined in Sec 4.3.3. At the end of the iterations that may run up to 10 times if the labelings do not stabilize earlier, the pairs labeled S are output as identified solutions (Line 13).

Time Complexity: Let n denote $|\mathcal{C}|$, and the number of unique words in each problem and reply post be a and b respectively. We will denote the vocabulary size of problem posts as A and that of reply posts as B. Learning of the language and translation models in each iteration costs $\mathcal{O}(nb + B)$ and $\mathcal{O}(k'(nab + AB))$ respectively (assuming the translation model learning runs for k' iterations). The E-step labeling and source estimation cost $\mathcal{O}(nab)$ each. For k iterations of our algorithm, this leads to an overall complexity of $\mathcal{O}(kk'(nab + AB))$.

5 Experimental Evaluation

We use a crawl of 140k threads from Apple Discussion forums[10]. Out of these, 300 threads (comprising 1440 posts) were randomly chosen and each post was manually tagged as either solution or non-solution by the authors of (Catherine et al., 2013) (who were kind enough to share the data with us) with an inter-annotator agreement[11] of 0.71. On an average, 40% of replies in each thread and 77% of first replies were seen to be solutions,

[10] http://discussions.apple.com

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

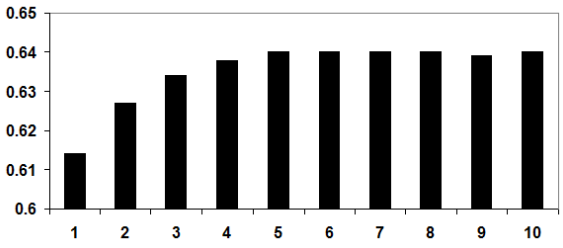

Figure 1: F% (Y) vs. #Iterations (X)

	ProblemWord, SolutionWord	$\mathcal{T}_S[p][s]$
$\mathcal{T}_S$	*network, guest*	0.0754
	connect, adaptor	0.0526
	wireless, adaptor	0.0526
	translat, shortcut	0.0492
	updat, rebuilt	0.0405
	SolutionWord	$\mathcal{S}_S[s]$
$\mathcal{S}_S$	*your*	0.0115
	try	0.0033
	router	0.0033
	see	0.0033
	password	0.0023

Table 4: Sample $\mathcal{T}_S$ and $\mathcal{S}_S$ Estimates

leading to an F-measure of 53% for our initialization heuristic. We use the F-measure[12] for solution identification, as the primary evaluation measure. While we vary the various parameters separately in order to evaluate the trends, we use a dataset of 800 threads (containing the 300 labeled threads) and set $\lambda = 0.5$ and $\tau = 0.4$ unless otherwise mentioned. Since we have only 300 labeled threads, accuracy measures are reported on those (like in (Catherine et al., 2013)). We pre-process the post data by stemming words (Porter, 1980).

5.1 Quality Evaluation

In this study, we compare the performance of our method under varying settings of λ against the only unsupervised approach for solution identification from literature, that from (Cong et al., 2008). We use an independent implementation of the technique using Kullback-Leibler Divergence (Kullback, 1997) as the similarity measure between posts; KL-Divergence was seen to perform best in the experiments reported in (Cong et al., 2008).

Table 3 illustrates the comparative performance

[12] http://en.wikipedia.org/wiki/F1_score

Technique		Precision	Recall	F-Measure
Unsupervised Graph Propagation (Cong et al., 2008)		29.7 %	55.6 %	**38.7 %**
Our Method with only Translation Models ($\lambda = 0.0$)		41.8 %	86.8 %	**56.5 %**
Our Method with only Language Models ($\lambda = 1.0$)		63.2 %	62.1 %	**62.6 %**
Our Method with Both Models ($\lambda = 0.5$)		61.3 %	66.9 %	**64.0 %**
Methods using **Supervision** (Catherine et al., 2013)	ANS CT	40.6 %	88.0 %	**55.6 %**
	ANS-ACK PCT	56.8 %	84.1 %	**67.8%**

Table 3: Quality Evaluation

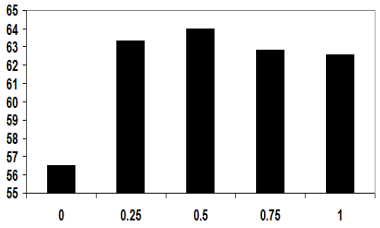

Figure 2: F% (Y) vs. λ (X)

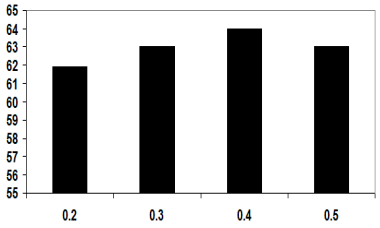

Figure 3: F% (Y) vs. τ (X)

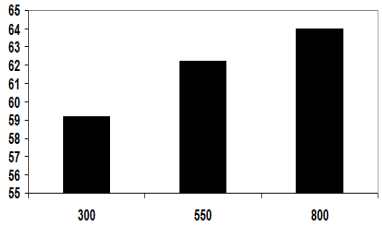

Figure 4: F% (Y) vs. #Threads (X)

on various quality metrics, of which F-Measure is typically considered most important. Our pure-LM[13] setting (i.e., $\lambda = 1$) was seen to perform up to 6 F-Measure points better than the pure-TM[14] setting (i.e., $\lambda = 0$), whereas the uniform mix is seen to be able to harness both to give a 1.4 point (i.e., 2.2%) improvement over the pure-LM case. The comparison with the approach from (Cong et al., 2008) illustrates that our method is very clearly the superior method for solution identification outperforming the former by large margins on all the evaluation measures, with the improvement on F-measure being more than 25 points.

Comparison wrt Methods from (Catherine et al., 2013): Table 3 also lists the performance of SVM-based methods from (Catherine et al., 2013) that use supervised information for solution identification, to help put the performance of our technique in perspective. Of the two methods therein, ANS CT is a more general method that uses two views (structural and lexical) of solutions which are then co-trained. ANS-ACK PCT is an enhanced method that requires author-id information and a means of classifying posts as acknowledgements (which is done using additional supervision); a post being acknowledged by the problem author is then used as a signal to enhance the solution-ness of a post. In the absence of author information (such as may be common in privacy-constrained domains such as medical forums) and extrinsic information to enable identify acknowledgements, ANS CT is the only technique available. Our technique is seen to outperform ANS CT by a respectable margin (8.6 F-measure points) while trailing behind the enhanced ANS-ACK PCT method with a reasonably narrow 3.8 F-measure point margin. Thus, our unsupervised method is seen to be a strong competitor even for techniques using supervision outlined in (Catherine et al., 2013), illustrating the effectiveness of LM and TM modeling of reply posts.

Across Iterations: For scenarios where computation is at a premium, it is useful to know how quickly the quality of solution identification stabilizes, so that the results can be collected after fewer iterations. Figure 1 plots the F-measure across iterations for the run with $\lambda = 0.5, \tau = 0.4$ setting, where the F-measure is seen to stabilize in as few as 4-5 iterations. Similar trends were observed for other runs as well, confirming that the run may be stopped as early as after the fourth iteration without considerable loss in quality.

Example Estimates from LMs and TMs: In order to understand the behavior of the statistical models, we took the highest 100 entries from both $\mathcal{S}_S$ and $\mathcal{T}_S$ and attempted to qualitatively evaluate semantics of the words (or word pairs) corresponding to those. Though the stemming made it hard to make sense of some entries, we present some of the understandable entries from among

[13]Language Model
[14]Translation Model

the top-100 in Table 4. The first three entries from $\mathcal{T}_S$ deal with connection issues for which *adaptor* or *guest account* related solutions are proposed, whereas the remaining have something to do with the *mac translator app* and *rebuilding* libraries after an *update*. The top words from $\mathcal{S}_S$ include imperative words and words from solutions to common issues that include actions pertaining to the *router* or *password*.

5.2 Varying Parameter Settings

We now analyse the performance of our approach against varying parameter settings. In particular, we vary λ and τ values and the dataset size, and experiment with some initialization variations.

Varying λ: λ is the weighting parameter that indicates the fraction of weight assigned to LMs (vis-a-vis TMs). As may be seen from Figure 2, the quality of the results as measured by the F-measure is seen to peak around the middle (i.e., $\lambda = 0.5$), and decline slowly towards either extreme, with a sharp decline at $\lambda = 0$ (i.e., pure-TM setting). This indicates that a uniform mix is favorable; however, if one were to choose only one type of model, usage of LMs is seen to be preferable than TMs.

Varying τ: τ is directly related to the extent of pruning of TMs, in the regularization operation; all values in the alignment vector $\leq \tau$ are pruned. Thus, each problem word is roughly allowed to be aligned with at most $\sim \frac{1}{\tau}$ solution words. The trends from Figure 3 suggests that allowing a problem word to be aligned to up to 2.5 solution words (i.e., $\tau = 0.4$) is seen to yield the best performance though the quality decline is graceful towards either side of the $[0.1, 0.5]$ range.

Varying Data Size: Though more data always tends to be beneficial since statistical models benefit from redundancy, the marginal utility of additional data drops to very small levels beyond a point; we are interested in the amount of data beyond which the quality of solution identification flattens out. Figure 4 suggests that there is a sharp improvement in quality while increasing the amount of data from 300 threads (i.e., 1440 (p, r) pairs) to 550 (2454 pairs), whereas the increment is smaller when adding another 250 pairs (total of 3400 pairs). Beyond 800 threads, the F-measure was seen to flatten out rapidly and stabilize at $\sim 64\%$.

Initialization: In Apple discussion forums, posts by Apple employees that are labeled with the *Apple employees* tag (approximately $\sim 7\%$ of posts in our dataset) tend to be solutions. So are posts that are marked *Helpful* ($\sim 3\%$ of posts) by other users. Being specific to Apple forums, we did not use them for initialization in experiments so far with the intent of keeping the technique generic. However, when such posts are initialized as solutions (in addition to first replies as we did earlier), the F-score for solution identification for our technique was seen to improve slightly, to 64.5% (from 64%). Thus, our technique is able to exploit any extra solution identifying structural features that are available.

6 Conclusions and Future Work

We considered the problem of unsupervised solution post identification from discussion forum threads. Towards identifying solutions to the problem posed in the initial post, we proposed the usage of a hitherto unexplored textual feature for the solution identification problem; that of lexical correlations between problems and solutions. We model and harness lexical correlations using translation models, in the company of unigram language models that are used to characterize reply posts, and formulate a clustering-based EM approach for solution identification. We show that our technique is able to effectively identify solutions using just one non-content based feature, the post position, whereas previous techniques in literature have depended heavily on structural features (that are not always available in many forums) and supervised information. Our technique is seen to outperform the sole unsupervised solution identification technique in literature, by a large margin; further, our method is even seen to be competitive to recent methods that use supervision, beating one of them comfortably, and trailing another by a narrow margin. In short, our empirical analysis illustrates the superior performance and establishes our method as the method of choice for unsupervised solution identification.

Exploration into the usage of translation models to aid other operations in discussion forums such as proactive word suggestions for solution authoring would be interesting direction for follow-up work. Discovery of problem-solution pairs in cases where the problem post is not known beforehand, would be a challenging problem to address.

References

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Rose Catherine, Amit Singh, Rashmi Gangadharaiah, Dinesh Raghu, and Karthik Visweswariah. 2012. Does similarity matter? the case of answer extraction from technical discussion forums. In *COLING (Posters)*, pages 175–184.

Rose Catherine, Rashmi Gangadharaiah, Karthik Visweswariah, and Dinesh Raghu. 2013. Semi-supervised answer extraction from discussion forums. In *IJCNLP*.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474. ACM.

P. Deepak, Karthik Visweswariah, Nirmalie Wiratunga, and Sadiq Sani. 2012. Two-part segmentation of text documents. In *CIKM*, pages 793–802.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Shilin Ding, Gao Cong, Chin-Yew Lin, and Xianyan Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *ACL*.

Ankur Gandhe, Dinesh Raghu, and Rose Catherine. 2012. Domain adaptive answer extraction for discussion boards. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 501–502. ACM.

Liangjie Hong and Brian D Davison. 2009. A classification-based approach to question answering in discussion boards. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 171–178. ACM.

Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and linking web forum posts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 192–202. Association for Computational Linguistics.

Janet L Kolodner. 1992. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34.

Solomon Kullback. 1997. *Information theory and statistics*. Courier Dover Publications.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137.

Zhonghua Qu and Yang Liu. 2011. Finding problem solving threads in online forum. In *IJCNLP*, pages 1413–1417.

Jangwon Seo, W Bruce Croft, and David A Smith. 2009. Online community search using thread structure. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1907–1910. ACM.

Amit Singh. 2012. Entity based q&a retrieval. In *EMNLP-CoNLL*, pages 1266–1277.

Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. 2008. Retrieval models for question and answer archives. In *SIGIR*, pages 475–482.