Complexity Guided Noise Filtering in QA Repositories

K. V. S. Dileep,[1] Swapnil Hingmire,[2] Sutanu Chakraborti[1]
[1]Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai - 600036, India
[2]Tata Research Development and Design Centre, Pune - 411013, India

Abstract

Filtering out noisy sentences of an answer which are irrelevant to the question being asked increases the utility and reuse of a Question-Answer (QA) repository. Filtering such sentences might be difficult for traditional supervised classification methods due to the extensive labelling efforts involved. In this paper, we propose a semi-supervised learning approach, where we first infer a set of topics on the corpus using Latent Dirichlet Allocation (LDA). We label the topics automatically using a small labelled set and use them for classifying an unseen sentence as useful or noisy. We performed the experiments on a real-life help desk dataset and find that the results are comparable to other methods in semi-supervised learning.

Introduction

The advent of Collaborative Question Answering (CQA) systems like Yahoo! Answers (http://answers.yahoo.com) and Quora (https://www.quora.com) have led to the creation of large archives of question-answer (QA) pairs. These QA archives have high reuse value in answering new queries which are similar to previously asked questions. Similarly, companies also generate a lot of help-desk queries and solutions which are rich sources of information that can be harnessed by powerful tools to help people solve problems.

However, an important drawback of all user generated content (UGC) is that the information is plagued with problems of noise. ‘Noise’ here not only means text with no meaning or syntactic errors like spelling mistakes, but also parts of the answer that do not add any value while answering the question and hence have low reuse value. Given the huge size of these repositories, manual filtering of noise is extremely difficult and this warrants an automated approach to isolate the parts of a solution irrelevant to a given problem.

For the scope of the problem we consider the noise to be of two types:

- Standalone noise: Noisy sentences that can be categorized as noise irrespective of the question being answered. These sentences are general filler sentences that do not add value. Example: Q: I am working in XYZ domain: i wanted to reset my ABC domain password. So please reset the password for ABC domain at the earliest A: ABC Domain password reset. Note: Please change the Password after 24 hours using the link below. *Please provide your valuable feedback before closing the ticket so that we can enhance the quality of our service.*
- Context-sensitive noise: These sentences are noisy w.r.t the question being asked. In the context of another question, these sentences might be labelled as useful. Example: Q: How to activate internet on mobile? A: Go to Settings and then go to Data Usage and check mobile data. *Special offers available on purchase of a new SIM. Call our toll free number for further details on special data packs available.*

In this paper, we consider the sub-problem of classifying sentences present in the answer into useful and noisy (irrelevant) sentences. A sentence is useful if it contains actions that solves the user’s issue; else it is considered noise. For effective sentence classification in a fully supervised setting, we require considerable data labelling effort and domain knowledge. Additionally, the labelled data might be insufficient for training, considering the volume of data available. In such situations it is useful to consider a topic modelling approach, where we model the corpus as mixtures of topics and label the topics instead of sentences. In our approach, we use a relatively limited amount of labelled data and use the sentence labels and word co-occurrences to label topics. This approach has a lower knowledge acquisition overhead compared to traditional supervised methods. Our experiments demonstrate that our procedure, while requiring less supervision, is better than most of other semi-supervised approaches w.r.t. classification effectiveness.

Related Work

Semi-supervised learning involves partial information being provided to the learning algorithm, instead of completely labelled data. Partial information can be provided through - labelling few documents, features or topics. Nigam et al.(Nigam et al. 2000) (EM LU) defined an algorithm for text classification from labelled and unlabelled documents based on the combination of Expectation-Maximization (EM) and a naive Bayes classifier. McCallum et al. (Mccallum and Nigam 1999) (EM Keywords) described an approach to build a classifier by using keywords on an unlabelled dataset. In the keyword based approaches, the key problem is to find the right sets of keywords for each class. One solution to this problem is to ask a user to provide a list of keywords, but it is a difficult task for the user to analyse the dataset and provide a complete list of keywords.

Hingmire et al. (Hingmire et al. 2012) (LDA-ML) uses LDA to obtain an initial set of topics for the documents. Then, experts are asked to label the topics directly as a useful topic or noisy topic. In this approach, all the labelling is done at the topic level. No labelled data is used for training. But labelling the topics directly may be difficult for humans. Our approach, in contrast, uses labelled information of sentences and transfers it to topics.

A variant of LDA called Supervised LDA (sLDA) which is completely supervised has been proposed by McAuliffe et al. (Mcauliffe and Blei 2008). sLDA works on labelled data and the goal is to infer topics that are able to predict the labels well. While sLDA directly learns the topics from the labelled data that are more likely to predict the label, our method learns topics from both labelled and unlabelled data and then use labelled information to filter topics.

Proposed approach

Topics are an abstract representation of documents than bag-of-words with lesser dimensionality. We obtain topics for our approach through Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003). With LDA, we represent each sentence in the dataset as a probability distribution of topics and topics are represented as probability distribution of words. Since LDA is an unsupervised learning algorithm, we do not know beforehand which topics help in discriminating the noisy sentences from the useful ones. To measure the discriminative power of a topic, we use the concept of alignment. Alignment measures the degree to which similar sentences have similar labels in our dataset. We use local alignment as defined in (Massie et al. 2007).

Topic Filtering

We perform the topic filtering step to estimate the utility of a topic in classification. From an initial set of topics, we try to arrive at a smaller subset of topics which are useful for classification in a greedy manner. We have a training set consisting of labelled and unlabelled sentences and an unseen test set. We run LDA with an initial number of topics on the entire training set - both labelled and unlabelled sentences. We now rank the topics based on the following measures:-

- **Coverage:** This measure represents the average probability of the topic in the entire document collection. Let $\boldsymbol{\theta}_i =< \theta_i^0, ..., \theta_i^{K-1} >$ be the sentence-topic probability distribution for sentence i. For the k^{th} topic, calculate topic coverage as $Coverage_k = \sum_{i=0}^{N-1} \theta_i^k$.
- **DiscGain:** This measure represents the discriminative power of a topic. It estimates the overall gain/loss in alignment on representing the document collection with addition of a new topic to existing topic set. A negative value indicates that the new topic may be unsuitable for classification.

These measures have a trade-off involved - Topics with high coverage may not be discriminative enough, while highly discriminative topics might not have ample coverage of the dataset. Thus, we calculate the final score of each topic ($tScore$) as

$$tScore = \delta \times Coverage + (1 - \delta) \times DiscGain. \quad (1)$$

The weight δ used in calculating $tScore$ is chosen through cross-validation. Topics with negative $tScore$ are dropped and not used for representing the sentences. We run LDA on the entire dataset, but calculate $tScore$ for a given topic only w.r.t the labelled subset of data.

Topic Labelling

Since a small subset of the sentence corpus is labelled, we can use the discriminating terms in the corpus to label the topics. We obtain the discriminating terms through a standard feature selection algorithm like chi-square (Yang and Pedersen 1997) that also gives the strength of the features. The label of a topic is dependent on the probability of a labelled word in the topic and strength of the word in determining the class. Highly probable words which are strong features (i.e. with high chi-squared statistic value) have a greater weight while labelling the topic. Instead of a hard assignment of a class to topic, we perform a soft assignment to both the classes as defined by $s_l^k = \sum_{i=0}^{m-1} \phi_i^k \times \chi_i^l$ where s_l^k is the soft label score for a label $l \in \{useful, noisy\}$ for k^{th} topic, ϕ_i^k be the topic-word probability distribution for k^{th} topic and i^{th} word. χ_i^l is chi squared statistic of i^{th} word and label l.

Along with soft labels of the topic, we also estimate how much we can trust these assigned labels. We refer to this measure as $confidence$ w.r.t. a given topic. It is defined as $Conf_k = \frac{\sum_{i \in \mathcal{M}} \phi_i^k}{\sum_{i=0}^{n-1} \phi_i^k}$ where $\mathcal{M}$ is set of indices of most probable words marked with a label in the topic. Higher the confidence, higher the robustness of the assigned labels.

Sentence Classification

Once we have filtered and labelled the topics, the next step is to label the unseen test data. We infer the topics of the new sentences using the LDA model inferred earlier. The class label assignment for a topic $t \in \mathcal{T}$ with index k depends on

- The probability of the topic in generating the sentence as given by LDA θ_i^k
- The soft labels for the given topic s_l^k
- The confidence factor $Conf_k$ estimated during labelling

$$\omega_i^l = \frac{\sum_{k=0}^{T-1} \theta_i^k \times Conf_k \times s_l^k}{\sum_{j \in \mathcal{L}} \sum_{k=0}^{T-1} \theta_i^k \times Conf_k \times s_j^k} \quad (2)$$

where $\mathcal{L} = \{useful, noisy\}$ refers to class label set, ω_i^l refers to weight of label l for sentence i, k refers to the index of a topic in a list of topics $\mathcal{T}$, θ_i^k refers to proportion of the presence of k^{th} topic in i^{th} test sentence, s_l^k is the proportion of the class label l in the soft labelling of k^{th} topic and $Conf_k$ represents the confidence factor. We choose $L = \arg\max_{l \in \mathcal{L}}(\omega_i^l)$ as the final class label.

Modelling the noise

As discussed earlier, there are two kinds of noise - standalone and context-sensitive. Modelling all instances as one type of noise might adversely affect the performance of other noise instances. Hence, for an incoming test instance if we can model which type of noise it is, we can make a better classification. For standalone-type noise, we consider only the solution and corresponding label for classification. For context-sensitive noise, we augment the problem part too with the solution. This is done to model the fact that the solution is noisy w.r.t the problem. We perform training on both the models, and choose the right representation for the incoming test instance. The right representation is chosen by calculating the expected alignment for each representation. Expected Alignment for each representation $z \in \mathcal{Z}$ (only two representations) is calculated as $EA_z = \frac{\sum_{i=1}^{b} w_{iz} \times locAl_{iz}}{\sum_{i=1}^{b} w_{iz}}$, where $locAl_{iz}$ is the alignment of document $\mathbf{s_{iz}}$ (Sentence $\mathbf{s_i}$ in representation z)in training data and w_{iz} its corresponding similarity to $\mathbf{u_{iz}}$. Choose $r = \arg\max_{z \in \mathcal{Z}}(EA_z)$. We use representation r for $\mathbf{u_i}$ to generate label.

Experiments and Results

In this section, we evaluate our proposed approach - LDA with Topic Filtering and Automatic Labelling (LDA-TFAL) with other semi-supervised approaches and compare its performance with a supervised method based on topic modelling called Supervised LDA (Mcauliffe and Blei 2008).

Dataset

The dataset used for our experiments was collected from the help desk of a reputed IT company in India. It is made from tickets related to webmail and Lotus Notes applications. Initially, we collected all the problems and solutions of the help desk for these applications. We created a corpus of 996 randomly selected sentences along with problems. These sentences were labelled as either "useful" or "noisy" by two annotators. The Kappa statistic for inter-annotator agreement was found to be 0.8. We represent our dataset as $\mathcal{D} = d_1 \cup d_2$ ($d_1 \cap d_2 =$ NULL) where, d_1 is the set of all useful sentences ($|d_1| = 685$) and d_2 is the set of all noisy sentences ($|d_2| = 311$). Basic preprocessing, i.e. removing the stops words and the non-ascii characters has been done. We evaluated effectiveness of our approach by computing the Macro F_1-measure. F_1 of class $d_i \in \mathcal{D}$ is harmonic mean of the precision and recall values of class d_i. Macro F_1 measure is the average of F_1 of each class which gives overall effectiveness of a classifier.

Experiment Settings

First, we look at the validity of our hypothesis of modelling noise as standalone and context-sensitive, and that each test instance must be labelled by choosing the representation where the neighbourhood is more 'aligned'. We model all test instances as standalone noise (only the solution components) and as context-sensitive noise (problem and solution together) and compare with the case where we choose the best representation for each instance. The results are shown in Table 1. The results show that the dataset has dominantly standalone noise instances. Indiscriminately applying the same model to all instances might lead to decrease in performance. Thus, choosing the right representation based on alignment estimates is the right way to go.

Model	Macro F_1
Standalone	0.81
Context-Sensitive	0.73
Representation Choice	0.83

Table 1: Comparison of various noise models on the dataset for LDA-TFAL method.

We compared the effectiveness of our method in comparison to a supervised topic modelling method - Supervised LDA (sLDA) (Mcauliffe and Blei 2008). We took 70% of the data as training and 30% of the data as test, keeping the overall class distribution same as the original data. We partition the training data into six approximately equal portions through stratified sampling. We start the experiment with one labelled portion and rest of the data as unlabelled. sLDA is trained on the labelled portion while LDA-TFAL is trained on the entire data with the labelled portion used for labelling the topics. We also perform sLDA with bootstrapping (sLDABoot), where we use the supervised topic model learned on labelled data to label the unlabelled portion of training data. We also perform a bootstrapped version of our method, LDA-TFALB, where we first learn the labels of unlabelled data from labelled data, and then label the topics. We relearn a supervised topic model on the entire training data which is now labelled. With models trained using LDA-TFAL, sLDA, and sLDABoot we infer labels on the test data.

We used Mallet[1] to run LDA on the corpus and set the various parameters using cross validation. We chose the weighting parameter δ as 0.6 also by experimentation on validation set. We observed that around 2-3 topics were removed through the filtering process on an average. We chose around 100 discriminating terms for automatic labelling using chi-square feature selection method (Yang and Pedersen 1997). We have chosen these parameters empirically. For sLDA (and sLDAB), we used the source code provided by Chong et al.[2]. The comparison of Macro F_1 of both LDA-TFAL, sLDA, sLDAB and LDA-TFALB with increasing percentage of labelled portion in training data is shown in Fig 1.

We can observe that when percentage of labelled portion is very small, our method LDA-TFAL and LDA-TFALB outperforms both the sLDA and sLDAB methods. When the percentage of labelled data increases, LDA-TFAL (and LDA-TFALB) reaches a plateau, while sLDA and sLDAB catch up and outperform our method. sLDAB catches up faster than sLDA, since it uses the unlabelled data also through bootstrapping. This tells us that when we have less labelled data, our method might be a better choice compared to a completely supervised method such as sLDA.

[1] http://mallet.cs.umass.edu/

[2] https://www.cs.cmu.edu/ chongw/slda/

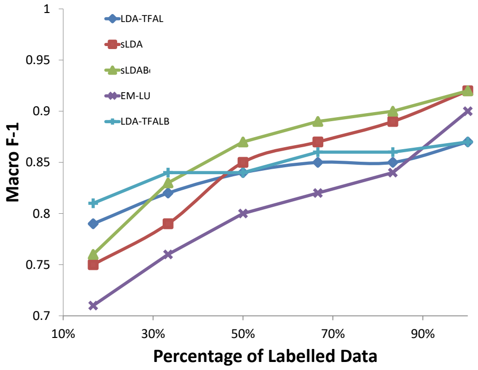

Figure 1: Comparison of classification performance of sLDA, sLDAB, LDA-TFAL, LDA-TFALB and EM-LU w.r.t percentage of labelled portion in training data.

This is because when there is insufficient labelled data, our method is able to leverage information present in unlabelled data to learn topics and label them. As the percentage of labelled data increases, sLDA has more data to refine the topic model.

Now, we compare the performance of our approach with other semi-supervised methods discussed in related work. We performed this experiment as follows. We partitioned our data into five parts with each part maintaining the same class distribution as the entire set. For each trial, we took three parts of training data - one part labelled and two parts unlabelled. The other two parts were set aside as test data. We performed five such trials with each trial using a different fold of the data. We took the results of these five trials - precision and recall at each trial and report their average. We take only 1/5th of data as labelled data to demonstrate that with a less labelling effort at the sentence level that is transferred to the topic level, we can get a good performance compared to other semi-supervised methods.

Algorithm	Useful class			Noisy Class			Macro
	P	R	F_1	P	R	F_1	F_1
LDA-TFAL	0.89	0.85	0.87	0.79	0.80	0.79	**0.83**
LDA-AL	0.83	0.77	0.80	0.73	0.75	0.74	**0.77**
LDA-ML	0.92	0.90	0.91	0.77	0.79	0.78	**0.85**
EM LU	0.88	0.89	0.88	0.75	0.73	0.70	**0.79**
EM key	0.86	0.90	0.87	0.78	0.69	0.73	**0.80**

Table 2: Experimental results of sentence filtering. We report the average across five trials for each of the methods.

We present our various results in Table 2. We found that individually, the methods perform better with choosing the model than applying same model for all instances just as in Table 1. We observe that our method performs better compared to the EM methods and falls a little short of LDA-ML, where topics are manually labelled. For LDA-ML, all topics are labelled by experts definitely as one class or the other. In our method, some topics contribute very little in the labelling process, since the highly discriminating words might have low probability. We don't compare our result with standard classification methods since the training data that is labelled is quite limited and the corresponding classifier might be insufficiently trained (as shown earlier during our comparison with sLDA). For each trial the number of topics vary, since there is a different labelled set.

Conclusion & Discussion

We have proposed a method of automatically filtering and labelling a few topics obtained through LDA using limited amount of labelled data and demonstrated its effectiveness on the problem of sentence filtering. We have also looked at our method in comparison to a supervised topic modelling method and showed that our method works better in case of availability of less labelled data. This work also complements the work done in Hingmire et al. (Hingmire et al. 2012) where the authors try to perform sentence filtering by soliciting labels on the topics directly. Labelling topics directly might become difficult for even an expert because he looks at the words of topics in isolation and does not have access to the sentences which provide the context. We also show that selectively choosing the representation based on the test instance is better when one representation adversely affects performance on other type of instances. A promising direction of future work would be to investigate how to select the sentences carefully so that with minimum labelling we can achieve good performance.

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research* 3:993–1022.

Hingmire, S.; Chougule, S.; Palshikar, G.; and Chakraborti., S. 2012. Almost unsupervised document content filtering using topic models. In *Proceedings of the Workshop on Analytics for Noisy unstructured text Data (AND 2012), 24th International Conference on Computational Linguistics*.

Massie, S.; Wiratunga, N.; Craw, S.; Donati, A.; and Vicari, E. 2007. From Anomaly Reports to Cases. In Weber, R. O., and Richter, M. M., eds., *Case-Based Reasoning Research and Development*, volume 4626 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 359–373.

Mcauliffe, J. D., and Blei, D. M. 2008. Supervised topic models. In Platt, J.; Koller, D.; Singer, Y.; and Roweis, S., eds., *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc. 121–128.

Mccallum, A., and Nigam, K. 1999. Text classification by bootstrapping with keywords, em and shrinkage. In *ACL-99 Workshop for Unsupervised Learning in Natural Language Processing*, 52–58.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning - Special issue on information retrieval* 39(2-3).

Yang, Y., and Pedersen, J. O. 1997. A comparative study on feature selection in text categorization. ICML '97, 412–420. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.