A Knowledge Based Approach for Tackling Mislabeled Multi-class Big Social Data

Minyi Guo[1], Yi Liu[1], Jie Li[1], Huakang Li[2], and Bei Xu[2]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
guo-my@cs.sjtu.edu.cn
[2] School of Computer Science & School of Software,
Nanjing University of Posts and Telecommunications,
huakanglee@njupt.edu.cn

Abstract. The performance of classification models extremely relies on the quality of training data. However, label imperfection is an inherent fault of training data, which is impossible manually handled in big data environment. Various methods have been proposed to remove label noises in order to improve classification quality, with the side effect of cutting down data bulk. In this paper, we propose a knowledge based approach for tackling mislabeled multi-class big data, in which knowledge graph technique is combined with other data correction method to perceive and correct the error labels in big data. The knowledge graph is built with the medical concepts extracted from online health consulting and medical guidance. Experimental results show our knowledge graph based approach can effectively improve data quality and classification accuracy. Furthermore, this approach can be applied in other data mining tasks requiring deep understanding.

Keywords: #eswc2014Li, label imperfection, knowledge graph, label correction, classification

1 Introduction

For machine learning research, many researchers focus on improving learning algorithms with least learning bias, thus the data quality has become the crucial issue when it is given to a certain machine learning algorithm. Unfortunately, real world data inevitably contains unexpected noises (i.e. label errors) which can disturb the performance of classification in multiple aspects like accuracy, modeling time and computing complexity. It proves that classification accuracies almost decline linearly with the increase of noise level [1].

Most label errors in training data come from data entry errors, transmit errors and subjectivity of taggers and so on. Data entry errors in large dataset are severe and common. The noise level is usually around 5% or more [1]. Furthermore, it seems difficult to avoid or even to cut down on the errors because there are no standards or specifications dealing with data entry errors. Transmission errors

take place in communication breakdown. Therefore, in order to increase the accuracy, most of training data are labeled manually even if the people are very subjective because of the knowledge limitation in specific domains. Even experts and professionals are not absolutely confident about their labeling. Therefore, the necessity of developing methods to remove or correct label errors is self-evident.

Many learning algorithms made label noised treatment mechanisms. For example, pruning in decision tree algorithm can avoid over-fitting caused by noises [2]. Still, when noise level is high, learning algorithms are not able to effectively. Other methods try to handle the noises in data before classification, including filtering noises and correcting noises.

This paper proposed an approach based on knowledge graph technique to perceive and correct label errors in big data environment. Knowledge graph is a concept proposed by Google[3] for its search engine and other applications, whose kernel is utilizing ontology to simulate entities and relationships in the real world to help machine understand the world intelligently. The usage of knowledge graph enable machines to better understand text documents [3]. Therefore we introduce this concept in noise correction to better perceive the nature conditions. We use big social data collected from medical Q&A web sites to validate our approach for tackling label imperfection. Medical Q&A system serves for online health consulting and medical guidance. A study reports 83% of Internet users in the U.S. seek health information online [4] and health care system are playing a much more essential role in the recent life [5].

Our approach implements the knowledge graph on a label correction method raised by Teng et al. [6]. Concretely, Naive Bayes classifier is utilized to recognize and modify the error labels of training data. After label modification, the noise level has proven to decline dramatically than before. Then we use the modified data to construct classifier for classification rather than correction, and the accuracy has improved than before. The main contributions of this paper are outlined as follows:

- We build a knowledge graph base containing medical entities such as diseases entities, symptom entities, medicine entities and their relationships from large scale of Q&A healthcare web sites, using several knowledge extraction techniques.
- We validate the effect of knowledge graph in tackling label imperfection problem comparing with other approaches. Our approach is more effective than other ways on improving classification quality and data quality.
- Our approach can be used for a relatively high noise level and still achieve satisfying performance.

This paper is organized as follows. Section 2 reviews the most related works in respects of label errors handling. Section 3 presents our approach to construct knowledge graph base. Section 4 describes polishing and our knowledge graph based combined approach. Section 5 describes the experimental performance and measures the affection of depth of knowledge as well. Finally, we conclude and discuss the possible directions of future works in Section 6.

[3] http://www.google.com/insidesearch/features/search/knowledge.html

2 Related Work

Over the course of the past 20 years, solving the problem of noises in the data has been the considerable attention in the field of machine learning and data mining. Most of learning algorithms developed mechanisms to diminish the impact that noises bring to the classification performance. Pruning in a decision tree is used to avoid overfitting caused by noise. Wilson et al. [7, 8] applied several instance-pruning techniques which can remove noise from the training set and reduce the storage consumption. However, the performance of these learning algorithms becomes very bad when the noise level is too high, and classification accuracy declines almost linearly with the rise of the noise level [1].

As long as the noise exists in training data, the classification quality will be affected severely. Thus, some approaches use filtering mechanisms to identify and filter the noise examples before feeding them to the classifier. Wilson et al. [9] attempted to filter the noise examples by using a 3-NN classifier and apply 1-NN classifier on the filtered data. Aha et al. [10] proposed IB3(a version of instance-based learning algorithm) to remove noise with lower updating costs and lower storage requirements. Brodley et al. [11, 12] used a set of learning algorithms to construct classifiers as filters to dataset before feeding it to classifier and achieved to significantly improvement for noise level up to 30%.

However, filtering noises enhances data quality at the cost of decreasing the amount of data retained for training. It also seems petty and inappropriate to discard error label data especially when the training data is difficult to re-collect such as historical data [13]. Correcting the label error instead of simply filtering them is a better approach that accomplishes both data quality and data amount. Zeng et al. [6] proposed a method called ADE (automatic data enhancement), which can correct label errors through numbers of iterations using multi-layer neural networks trained by back propagation in the basic framework. Teng et al. [13, 14] introduced a noise correction mechanism called polishing and correct noises both in classes and attributes. Teng also compared polishing with filtering and traditional approach of avoiding overfitting, and proved noise correction recovers information not available with the other two approaches [14, 15]. Since we apply polishing as our basic method, more detailed description about polishing will be presented in Section 4.1.

The approaches discussed above contain the following limitations: (i) Some use filtering which may decrease the bulk of data. (ii) Most of these approaches have no significant performance at a high noise level. (iii) Most of these works only measured the promotion that their approaches bring to classification performance, yet haven't measured the exact values of data quality promotion. Therefore, we propose an approach based on knowledge graph to tackle these limitations.

3 Knowledge Graph Building

3.1 Data Source

We use a big dataset over 1000GB collected from a Chinese medical social Q&A website[4] and Chinese Encyclopedia website Baidu Encyclopedia (BE)[5] to build a medical knowledge base. Figure 1 shows a glimpse of a few entities and relationships in the graph. The edge between a disease entity and a symptom entity implies the disease seems to have a lot of symptoms. For example, *gastritis* has *diarrhea* and *vomit* symptoms, and *fatigue* can be explained by *anemia* or *Parkinson*. There are 3 types of entities in the knowledge graph, and two entities of the same type cannot be connected directly. This assumption is justifiable. Because in the real world, two diseases are related since they share several common symptoms. Two medicines are related since they can be both employed to treat one disease. Their relationship is linked by other entities, not themselves directly.

Besides, the Q&A archives are used to establish training data sets applied for label correction. The Q&A archives contain nearly 20 million Q&A pairs in which every pair contains the question put forward by patients and the answer given by doctors and medical experts. The pair also contains departmental information about which hospital department the patient should seek help for. It's appropriate to use these data to validate our approach. We extract a training example from each Q&A pair. Features are extracted from patients' descriptions in questions, and departments are used as labels in the correction phase.

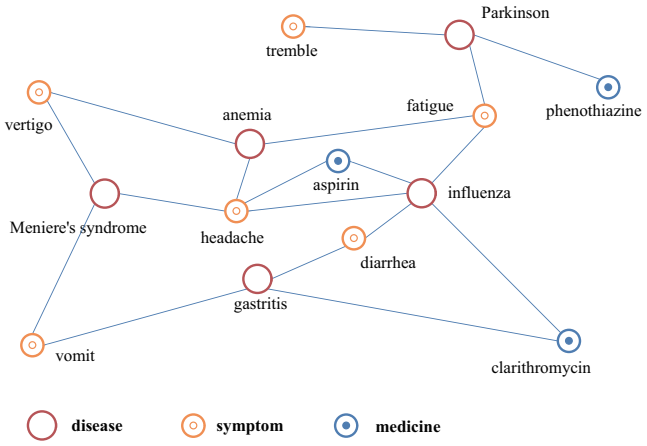

Fig. 1. A local structure of the medical knowledge graph

[4] http://www.120ask.com
[5] http://baike.baidu.com

3.2 Entities Extraction

To build the knowledge graph base, we extract disease entities, symptom entities and medicine entities. These are done by following steps:

- In the first phase, we use web crawling technique to acquire disease entities, medicine entities from BE. As BE pages are well structured and tagged, we adopt Maximum Entropy algorithm to classify these entities to broad categories. After sorting out these entities and their categories, we obtain a known entity set.
- In the second phase, we conclude linguistic patterns of entities and use these patterns to find more entities in the Q&A archives. Bootstrapping on syntactic patterns are frequently used to extract knowledge [3]. Chinese words are composed of characters, and affixes (prefixes and suffixes, contains one or more characters) usually have specific meaning about the type of words. For example, medicine words 'mizolastine', 'clemastine' and 'levocabastine' all share the same suffix 'stine', because they are similar in chemical composition. So we use prefixes and suffixes concluded from the known entities set to find more and more entities. After acquiring these new entities, we conduct artificial selection to discard entities which do not belong to the medical domain. Hence, we get a bigger set of entities than the first phase.
- Then we perform several iteration of the second phase and finally get a set of nearly 30,000 disease entities and 30,000 medicine entities.

Since most patients describe their symptoms orally and informally, symptoms cannot be extracted from encyclopedia web sites. We firstly use TFIDF [16] and IG(information gain) techniques [17] to find words and phrases that are more informative in the Q&A archives, and artificially select some symptom entities. Then we use bootstrapping to seek more and more symptom entities. Finally we obtain a set of nearly 4,000 symptom entities.

3.3 Relationship Extraction

In most of the existed knowledge bases such as Wikipedia[6], Freebase[7], YAGO[8], Wordnet[9], the relationships between entities or relationships between entities and their attributes are established manually by experts in related field. Our knowledge base contains a relatively big amount of entities and we don't have professional knowledge in medical taxonomy. Therefore we adopt a method to automatically extract relationships between entities from big data, whose details will be discussed in Section 4.2.

In our opinion, the entity that occurs simultaneously in one Q&A pair that has some relationships. We make an assumption that the more frequently entities

[6] http://www.wikipedia.org
[7] http://www.freebase.com
[8] http://www.mpi-inf.mpg.de/yago-naga/yago
[9] http://wordnet.princeton.edu

occur simultaneously in Q&A pair, the stronger relationships they have. Hence, we extract relationships between entities based on the $co-occurrencerate$ of entities. Details on $co-occurrencerate$ are discussed in Section 4.2.

4 Mislabel Correction

As we mentioned above, polishing proposed by Teng et al. [13, 14] proves to be quite well in mislabeled correction. The kernel of our approach is to adopt polishing as the basic method and use information from the established knowledge graph to adjust the weight of entity features in label correction phase. Since knowledge graph represents the relationships of entity features, it can be utilized to strengthen the more informative entity features and weaken the less informative entity features. We assume that the entity with more connection to other entities and greater co-occurrence rates with others plays the more important role in mislabeled correction. Thus, they should be endowed with more weight.

4.1 Polishing

The basic polishing algorithm comprises two phases: prediction and adjustment [14]. The prediction phase aims at finding candidate training examples that are suspected to control error labels, while the adjustment phase decides the final changes into the candidates. The polishing algorithm can predict and correct both attributes errors and label errors(i.e. class errors). In this paper, we use it to correct label errors.

In the prediction phase, a chosen learning algorithm performs K-fold cross validation. Teng et al. set K to be 10. The K-fold cross validation divides all the examples in K groups called folds, and constructs K classifiers each using K-1 folds as training set and the folding left out as the test set. If the K-fold cross validation algorithm predicts a label inconsistent with the original label, this sample will be added to suspected candidates.

In the adjustment phase, for each example of candidates set, K classifiers constructed in the prediction phase are used to predict labels of this example. If the predicted labels of K classifiers are identical and different from the original label, polishing judges the new label to be the right one and modifies the example using the new label.

4.2 Knowledge Graph

We define our knowledge graph to be a set of vertices $(v_1, v_2, \ldots, v_m)$ and edges $(e_1, e_2, \ldots, e_m)$. Each vertex represents an entity and each edge represents a direct relationship between two entities. Direct relationship means a strong connection between two entity vertices. For instance, a brief example of relationships of several entities have been shown in Fig. 1, *gastritis* has symptoms of *vomit* and *diarrhea*, so they are connected directly. And the relationship between $Meniere's\ syndrome$ and *gastritis* cannot be described, we only know they share some common symptoms, so their relationship is indirect.

We define *distance* as the shortest path length between two vertices. *distance* between any two vertices can be computed once the *length* of any edges is known. The *length* of edge is computed using the formula:

$$length(v_i, v_j) = \frac{1}{co - occurrence\ rate(v_i, v_j))} \tag{1}$$

$co - occurrence\ rate$ can measure closeness of two entity vertices if they have direct relationship. The smaller *length* is, the larger $co - occurrence\ rate$ is, meaning the relationship between two entity vertices is closer. The $co - occurrence\ rate$ is computed from the Q&A data according to the formula:

$$co - occurrence\ rate(v_i, v_j) = \frac{2 * n_{ij}}{n_i + n_j} \tag{2}$$

Here v_i, v_j represents any two entity vertices. n_{ij} represents the number of Q&A pairs in which v_i and v_j occur simultaneously, n_i defines the account of pairs in which v_i occurs, and n_j defines the number of pairs in which v_j occurs. Apparently the $co - occurrence\ rate$ is maximum value 1 if two entities always occur simultaneously in Q&A pair. If $co - occurrence\ rate$ is below a threshold M, we assume the two entity vertices have no direct relationship, thus no edge existing between them.

Also, we define *related degree* to measure relationship closeness between two vertices even when they are not directly connected in the knowledge graph (namely no edge between them).

$$related\ degree(v_i, v_j) = \frac{1}{distance(v_i, v_j)} \tag{3}$$

Obviously *related degree* is equivalent to $co - occurrence\ rate$ when there is an edge directly connecting two entity vertices. *distance* is computed using Dijkstra Shortest Path algorithm [18]. And we define $step(v_i, v_j)$ as the edge num of the shortest path between v_i and v_j. *step* measures the depth of knowledge we dig in the graph.

One advantage of knowledge graph is that we can extend or modify the graph once we grasp new knowledge through science researches. When we discover a new disease, we add it into the graph and connect it to other symptoms or medicines based on the information we know about it. And if the latest medical research shows some kind of medicine can help treat a disease, which hasn't been applied before, we can connect them and endow them some kind of relationship.

4.3 Weight Adjustment

Numerous feature weighting methods have been applied to classification and prove to have a promotive effect on classification accuracy. These methods include information gain (IG), term frequency-inverse document frequency (TFIDF), mutual information (MI), χ^2 statistic (CHI) [17]. Most of them depend on statistical analysis on training data to select and strengthen the informative features.

When applying these methods in label correction, the noise part of training data probably interferences the outcome when the noise level is relatively high. Therefore we use knowledge graph to adjust weights of entity features, because knowledge graph has several advantages as below:

- Knowledge graph technique is able to mine deep relationships among features, while traditional statistical methods simply analyze shallow relationships among features.
- Knowledge graph is similar to a real world model. It is more reasonable and precise to simulate relationships.
- The knowledge cannot only be extracted from corpora but also come from scientific knowledge and latest research, which makes the graph to be extensible and renewable.

Specifically, we compute the weights of entity features according to the formula:

$$\begin{aligned} weight(v_i) = & initial\ weight + \alpha \sum_{v_j \in V, v_j \neq v_i} related\ degree(v_i, v_j), \\ & \forall step(v_i, v_j) < MAXSTEP \end{aligned} \tag{4}$$

V is the vertices set in the graph and $MAXSTEP$ is defined as the depth of relationships we mine. We define *initial weight* to be 1, and α is the adjustment factor to control the impact of knowledge graph to feature weights. $MAXSTEP$ sets a limit to which vertices to be considered when computing the weight of a vertex, namely the analysis depth of knowledge graph. We believe the weight is more specific if the depth goes deeper. However, there is a tradeoff between analysis depth and computational complexity because the related vertices number is quite large when we analyze graph quite deeply. We will conduct experiments about the effect of knowledge depth on correction labels in the Section 4.2.

4.4 Combined Algorithm

Our approach combines polishing and weight adjustment by knowledge graph to correct noise labels in training examples. We use Multinomial Naive Bayes (MNB) classifier as the basic classifier in K-fold cross validation. We choose MNB because it proves to be both efficient and accurate for text classification tasks [19]. Still, MNB makes a poor assumption that features of examples are independent of others, which are clearly unreasonable in most real-world tasks. We adjust feature weights in MNB classifier according to knowledge graph to compensate for this assumption. Weights of entity features are calculated according to formula (4) and weights of other features are defined as 1. When corrupt training data is prepared, we adjust the weight of features in the training examples, and get the adjusted training data. Then we utilize this data to follow the same procedures for polishing in Section 4.1. We also set K to be 10

in the K-fold cross validation. Afterwards we can obtain data corrected by our combined approach. Experiments of our combined approach to medical Q&A data will be revealed in the following section. We will evaluate the effect of our approach on both classification accuracy and data quality promotion.

5 Experiment and Evaluation

This section provides empirical evidence that our knowledge graph based approach is effective in improving data quality and classification scores.

5.1 Data sets

Table 1. the format of Q&A pairs

description	answer	department
I'm 23 and my hands always shake...and it gets worse when I'm nervous...	There are many reasons for your shaky hands. It's hard to guess it...	neurology
I play badminton and when I use backhand serve, my hand tremble. My brachioradialis hurts too...	It may be caused by overexercise, I suggest you see a bone surgery doctor to ...	surgery

As we mentioned above, our data is extracted from a huge set of nearly 20 million medical Q&A pairs. The format of data is specified in Table 1, each example has a description text which patients depict about their circumstances and symptoms, and each example has a department label showing the department where this patient should be treated. The description text of Q&A pair is usually short, less than 200 characters. The whole data sets contain more than 10 departments, Table 2 shows the department names and their probability distribution. We use our approach to obtain and correct the error department labels in training examples. Since the corpus is in Chinese, we use several NLP methods specialized in handling Chinese text: tokenizing Chinese text and transfer traditional Chinese characters to Chinese simplified characters. Afterwards, we extracted approximately 200,000 features from the raw data. Finally, we get nearly 9,725,000 training instances.

In order to obtain the corrupt data, we artificially corrupt the data with random label noises. In the following subsections we will conduct our approach with different noise levels.

5.2 Evaluation Measures

As Teng et al. points out, there are two kinds of measurement methods to evaluation label correction [13]. One method aims at finding out to what degree the

Table 2. department labels and their distribution

department	distribution
obstetrics and gynaecology	26.6%
internal medicine	20.4%
surgery	11.3%
pediatrics	9.9%
dermatology	7.9%
ophthalmology and otorhinolaryngology	5.8%
neurology	5.5%
psychology	5.1%
traditional Chinese Medicine	3.1%
infectious diseases	1.9%
oncology	1.9%
plastic surgery	1.0%

label correction improves classification score, including accuracy, F1 score, F2 score etc. We choose accuracy as the measure metric to evaluate the classification quality improvement after label correction. The other method measures the data quality in a classification-independent way, considering we may want to put the corrected data in additional uses other than building classifiers. Unlike the Net Reduction and Correct Adjustment used by Teng [13] to measure reduction in attribute noises, we use different metrics to evaluate the data quality promotion. These metrics are *noise reduction rate*, *precision* and *recall*. As our approach and in polishing correct labels by the judgement of 10 classifier voters, the changes made to the examples are not always right. So these metrics are used to evaluate these changes. *noise reduction rate*(NRR) is defined in (5) and measures the noise level decrease after label correction. *precision* measures the percentage of right changes in the whole changes made by label correction approaches. *recall* measures the percentage of error labels which is actually corrected. It's obvious that *noise reduction rate* most intuitively reflects the data quality promotion.

$$NRR = noise\ level\ in\ origin\ data - noise\ level\ in\ corrected\ data \quad (5)$$

We use three methods: *Unpolishing*, *Polishing* and *Polishing* + KG in classification accuracy comparison. *Unpolishing* approach uses the unmodified corrupt data to build classifier. *Polishing* approach uses the data corrected by polishing method to build classifier. And *Polishing*+KG approach uses the data corrected by our approach to build classifier. All the three approaches are applied in accuracy comparison, and the latter two are applied in mislabeled reduction rate comparison. In addition, we set $MAXSTEP$ to 1 in *Polishing*+KG when compared with other two approaches.

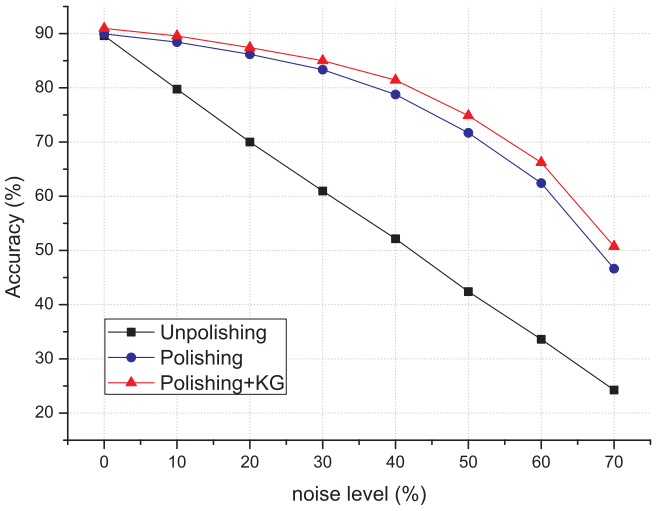

Fig. 2. A comparison accuracy on data by *Unpolishing*, *Polishing* and *Polishing* + *KG* on the medical Q&A data set

5.3 Classification Accuracy

We compare the classification accuracy on training data produced by three approaches mentioned above. For each approach, 10-fold cross validation is performed on data to obtain classification accuracy. In each trial, nine folds are used for training data to test the accuracy of the rest fold. The final accuracy is the average accuracy of 10 trials. Here we use cross validation to evaluate classification accuracy, different from label correction phase where cross validation is used to pick up candidates and construct classifiers as voters. We choose cross validation to validate accuracy because it can reduce the risk of overfitting on the test set.

Figure 2 shows the comparison of three approach on classification accuracy at different noise levels. For *Unpolishing* approach, accuracy declines almost linearly with the noise level increase. At most cases, the improvement of *Polishing* and *Polishing* + *KG* on *Unpolishing* is quite significant, the performance of *Polishing* is 10% - 30% higher than *Unpolishing*, while our approach *Polishing* + *KG* acquires accuracy usually 1% - 4% higher than the pure *Polishing*. We can see noise data cut down accuracy dramatically when no correction is conducted. *Polishing* corrects part of the error labels and provides a much higher accuracy. Furthermore, *Polishing* + *KG* approach mines the relationships between entity features and endows more weights to the more informative ones, so it achieves better accuracy score than *Polishing*. Particularly, at noise level of 0%, the improvements of *Polishing* and *Polishing* + *KG* are both not remarkable, *Polishing* is merely 0.3% higher than *Unpolishing*, and *Polish* + *KG* is 1.3% higher than *Unpolishing*, we believe *Polishing* + *KG*

also has effect on improving classification accuracy even when data is nearly noise-free.

5.4 Data Quality Promotion

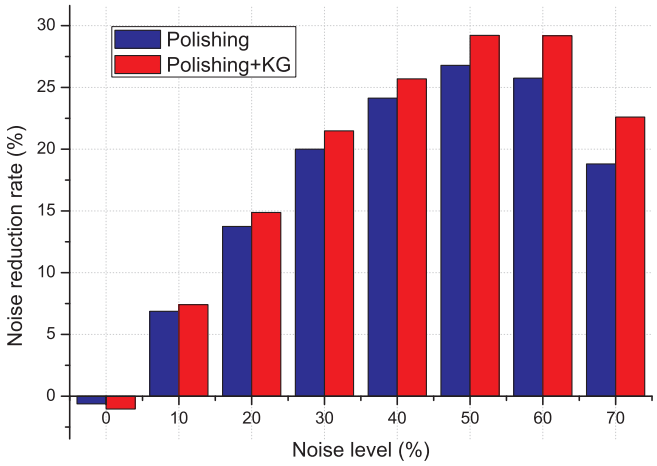

Fig. 3. A comparison of noise reduction rate by *Polishing* and *Polishing* + *KG* on the medical Q&A data set

We compare the classification-independent metrics to test data quality promotion by *Polishing* and *Polishing* + *KG* approach. When we artificially corrupt the data, we have made a mark to every example what is the real label of it. After label correction by two approaches, we check the precision, recall and noise reduction rate depending on these marks. We use noise reduction rate as the main metric on data quality promotion, while the other two help us to understand and explain the relevant promotion.

Figure 3 shows noise reduction rate by two approaches. The noise reduction rate of *Polishing* + *KG* is approximately 1% - 4% higher than *Polishing*. It seems odd that the noise reduction is negative at noise level of 0%, which means the noises increase after label correction. However, this phenomenon can be explained. At noise level of 0%, we assume data to be noise-free, while data can't be completely noise-free in real-world. So it is reasonable that *Polishing* and *Polishing* + *KG* has modified some labels which are quite possibly error labels. Generally speaking, it is shown that *Polishing* has enormous significance in data quality promotion and *Polishing* + *KG* achieves better performance on the basis of *Polishing*.

Figure 4 shows the precision and recall. We do not considerate precision and recall at noise level of 0% because it's meaningless. At most noise levels, precision of *Polishing* + *KG* is less than *Polishing*, however the recall of *Polishing* + *KG*

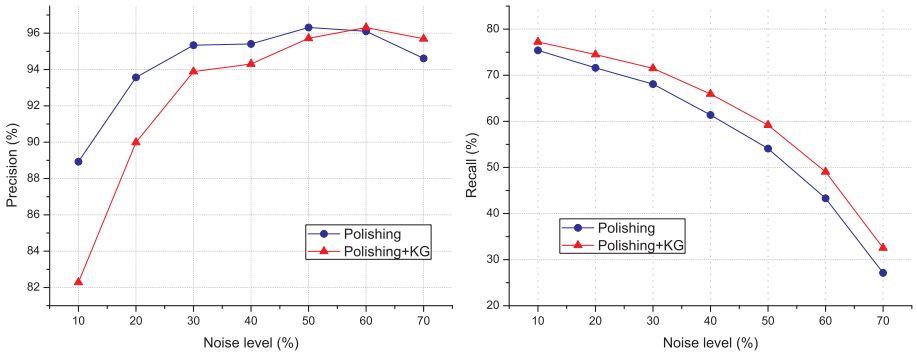

Fig. 4. A comparison of precision, recall by *Polishing* and *Polishing* + *KG* on the medical Q&A data set

is much higher than *Polishing*. Usually precision and recall have a contradictory relationship that precision decreases along when recall increases. So it's reasonable that *Polishing* + *KG* has a lower overall precision. When the noise level is quite higher, the precision and recall of *Polishing* + *KG* are both higher than *Polishing*. We assume this is caused by that knowledge diminishes the interference of noises, the effect is more remarkable when the noise level is higher.

5.5 Knowledge Depth Affection

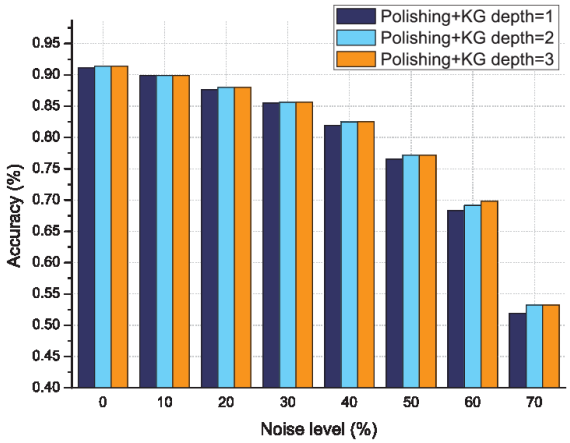

Fig. 5. Knowledge depth affection on accuracy

We conduct an experiment of how knowledge depth affects the results. According to (3), we adjust the entity weights by computing closeness of an entity to other entities. We believe the bigger $MAXSTEP$ is, the more precise weights will be generated. This thought is driven by that we get more information about something when we recognize it more deeply. Figure 5 shows the accurate comparison of different knowledge depth from 1 to 3. The accuracy improves 0-1.3% when knowledge depth grows from 1 to 2 at different noise levels, while the accuracy improvement is insignificant when depth grows from 2 to 3. When knowledge depth grows, the amount of relationships of one entity to others grows rapidly and more weights are endowed with the more informative ones. The results show deep knowledge perception can enhance classification performance.

6 Conclusion

In this paper, we present a knowledge graph based approach combined with polishing to handle label imperfection problem. This method is distinct from previous statistical methods in that it tries to recognize the data in a way similar to the real world. Experimental results demonstrate our approach has an impact on boosting classification performance and data quality. It can effectively correct mislabeled even under the circumstance of a quite high noise level of approximately 60%. Beside handling the noise data, the knowledge graph technique we used can be applied in feature selection in classification as well.

Our future work will be focused on ameliorating the graph by establishing more types of entities and more detailed relationships in it. More researches will be conducted to recognize data noises in a more human-like rather than machine-like approach. In addition, we shall apply our approach to other fields such as social networks and business data analysis.

7 acknowledgement

This work was supported by the NSFC (No. 61272099, 61261160502 and 61202025), Shanghai Excellent Academic Leaders Plan(No. 11XD1402900), the Program for Changjiang Scholars and Innovative Research Team in University of China (IRT1158, PCSIRT), the Scientific Innovation Act of STCSM(No.13511504200), Singapore NRF (CREATE E2S2), and the EU FP7 CLIMBER project (No. PIRSES-GA-2012-318939).

References

1. Zhu, X., Wu, X.: Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review **22**(3) (2004) 177–210
2. Quinlan, J.R.: Induction of decision trees. Machine learning **1**(1) (1986) 81–106
3. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: A probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM (2012) 481–492

4. Zhang, Y.: Contextualizing consumer health information searching: an analysis of questions in a social q&a community. In: Proceedings of the 1st ACM International Health Informatics Symposium, ACM (2010) 210–219
5. Kunz, H., Schaaf, T.: General and specific formalization approach for a balanced scorecard: An expert system with application in health care. Expert Systems with Applications **38**(3) (2011) 1947–1955
6. Zeng, X., Martinez, T.R.: An algorithm for correcting mislabeled data. Intelligent data analysis **5**(6) (2001) 491–502
7. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: ICML. Volume 97. (1997) 403–411
8. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine learning **38**(3) (2000) 257–286
9. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. Systems, Man and Cybernetics, IEEE Transactions on (3) (1972) 408–421
10. Aha, D.W., Kibler, D.F.: Noise-tolerant instance-based learning algorithms. In: IJCAI, Citeseer (1989) 794–799
11. Brodley, C.E., Friedl, M.A.: Identifying and eliminating mislabeled training instances. In: AAAI/IAAI, Vol. 1, Citeseer (1996) 799–805
12. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. arXiv preprint arXiv:1106.0219 (2011)
13. Teng, C.M.: Evaluating noise correction. In: PRICAI 2000 Topics in Artificial Intelligence. Springer (2000) 188–198
14. Teng, C.M.: Polishing blemishes: Issues in data correction. Intelligent Systems, IEEE **19**(2) (2004) 34–39
15. Teng, C.M.: A comparison of noise handling techniques. In: FLAIRS Conference. (2001) 269–273
16. Li, J., Zhang, K., et al.: Keyword extraction based on tf/idf for chinese news document. Wuhan University Journal of Natural Sciences **12**(5) (2007) 917–921
17. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML. Volume 97. (1997) 412–420
18. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1) (1959) 269–271
19. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. Volume 752., Citeseer (1998) 41–48