Enhancing Usability of Information Extraction Results with Textual Data Profiling

Jyi-Shane Liu and Yung-Wei Cheng
Department of Computer Science
National Chengchi University
64 Sec. 2 Chihnan Rd.,
Taipei 116, Taiwan
jsliu@cs.nccu.edu.tw

Abstract

Given a targeted subject and a text collection, information extraction techniques provide the capability to populate a database in which each record entry is a subject instance documented in the text collection. However, even with the state-of-the-art IE techniques, IE task results are expected to contain errors. Manual error detection and correction are labor intensive and time consuming. This validation cost remains a major obstacle to actual deployment of practical IE applications with high validity requirement. In this paper, we propose a string feature-based approach to textual data profiling and invalid data detection. The approach is based on the observation that values of an attribute in IE results are symbolic form variations of a concept in the IE task subject and may exhibit a certain congruity with some string features. We conducted experiments to verify that effective detection of IE invalid values can be achieved by using the surface-form string features.

1. Introduction

Most IE researches are concerned with producing better performance and facilitating development portability [2][5][1]. Even with the state-of-the-art IE techniques, IE task results are expected to contain errors. If the IE results are to be used for applications with data validity concerns, IE errors need to be detected and corrected before any value-added processing can be applied. For a typical IE application with tens of thousands, or even hundreds of thousands, extracted entities, manual error detection and correction are labor intensive and time consuming in checking IE results with original documents. This validation cost remains a major obstacle to actual deployment of practical IE applications. In some cases, the validation cost may be so high that IE techniques appear to be hardly superior to direct human work in which both extraction and validation may be performed at the same time. Unfortunately, issues of ensuring highly validated IE results with acceptable cost are little addressed in the IE community. If IE techniques are to be translated into adequately supported actual deployment, we must have some ways to reduce validation cost and ensure data quality of IE results.

Data cleansing, also called data cleaning, considers the problem of identifying and removing errors and inconsistencies in data sets [3]. Due to the wide range of possible data errors and inconsistencies, data cleansing involves a variety of research efforts to deal with invalid data, missing data, and duplicated data that occur in single source or when integrating multiple sources. One of the primary focuses is on data integrity analysis or data auditing. The goal is to analyze the actual instances of data to obtain indicative data characteristics or value patterns. The results of data analysis are then used to locate potential errors and inconsistencies based on anomaly and conflict detection. Two related approaches, data profiling and data mining, have been attempted for data analysis [4]. Data profiling focuses on individual attributes and derives information such as data type, length, value range, variance, value frequency, occurrence of null values, typical string pattern, etc. Data mining helps discover specific data relationships between several attributes, such as dependencies or domain-specific business rules. Both approaches provide indispensable tools for data cleansing tasks.

In this paper, we present a string feature-based approach to textual data profiling and invalid data detection. A set of generic string features is proposed to provide characteristic profiling for textual data. Each string feature describes a surface property of a string without concerning its literal meaning. Textual data are audited by examining their string features and locating potential errors with atypical feature values. We consider two complementary strategies to establish the classification standards. The first strategy relies on specifying attribute constraints with a-prior domain knowledge. The second strategy employs the concept of statistical majority. The approach was applied to detecting invalid data in IE results from the task domain of government personnel directives. Performance evaluation shows the approach is capable of effectively classifying data validity and anomaly. Finally, we discuss certain observations and issues from the experiments, followed by suggestions for future explorations.

2. Textual Data from Information Extraction

The goal of IE tasks is to scan through a collection of documents and fill in a table of values with extracted strings from relevant text. Each instance of the targeted subject (or event) appeared in the text is to be transformed to an entity in a database with attributes specified by the information template of the subject. The output of an IE task, therefore, is a database first populated with exclusively textual data representing groups of subject instances.

2.1. Error Analysis

In general, data cleansing approaches and methods are closely related to the types of errors handled by them. Effective data cleansing for IE results would need to identify what types of errors are considered. IE results are derived by automatic data acquisition from raw text with computer processes. Extraction errors are the primary cause of data anomalies. From the point of view of data acquisition, IE techniques select a subset of words from raw text and fill them into a set of attributes. Errors occur both in selecting words and in matching selected words to attributes.

We classify between four types of data anomalies in IE results.

●Missing values: IE techniques fail to identify the correspond-ing value for an attribute from the text. As a result, the attribute slot is left empty.

●Missing entities: IE techniques fail to recognize the subject entity entirely from the text. As a result, no corresponding entry is recorded.

●Invalid values: IE techniques make mistakes in segmenting sequences of words or in associating words to attributes. As a result, attribute values are invalid either at the syntactical or the semantic level.

●Duplicates: Duplicates are multiple entries representing the same subject entity. The occurrence of duplicates is usually not a mistake of IE techniques but is mostly due to repeated multiple descriptions of the same subject entity in text collection.

2.2. IE Data Anomaly Problem

Ideally, data cleansing for IE results should be able to detect and remove all four types of data errors. However, the textual nature of IE results presents much more obstacles for data analysis than numeric data. Among the four types of data anomalies, duplicates are easier to detect with information of key attributes. Missing values are mostly dependent on the availability of integrity constraint. If an attribute is known to be required, an empty slot provides a straightforward indication of a missing value. Otherwise, it is very hard to detect anything as missing when there is no information on its existence. The same applies to missing entities, which is not possible to detect by analyzing IE results alone.

Our primary focus will be on detecting invalid values. Given an initial textual database loaded by an IE task, we address the problem of analyzing the actual data content and detecting values that are either syntactically or semantically invalid. We consider a simplified database in the form of a table where each row is a subject entity and each column corresponds to an attribute. Each attribute may or may not associate with integrity constraints defined by a-prior knowledge of the task domain. The goal is to detect each and every invalid value in the textual database. Such a detection capability offers significant help to ensuring high-quality IE applications and is the beginning step towards more complete data cleansing with anomaly removal and correction.

3. Featured-based Detection on Invalid Values

We propose a set of generic string features that can be used to classify textual data into groups with different feature values. Each string feature describes a certain characteristics of a string. Ideally, invalid data will be exclusively separated out from normal data based on their unique string feature values. We observe that attribute values of different entities are only various instantiations of the same concept. Values in each column can be considered as the same class of textual string that shares certain characteristics. String features provide a basis to find the common characteristics of a class of textual strings. Groups with atypical string feature values indicate their possible anomaly.

3.1. String Features

A textual string, as an attribute value in IE results, is usually a word or a sequence of words that provides a unit of information pertaining to a subject entity. In linguistics, morphology concerns rules of word formation. A word has a form at the surface level and a meaning at the content level. Word meaning classification requires dictionary, which is subject dependent and does not work for entity name. On the other hand, word forms allow direct processing to determine its surface properties. Therefore, we propose to detect anomaly of attribute textual values based on surface forms of words. Our basic assumption is that certain surface properties of words would help provide differentiable characteristics between normal and abnormal values. Invalid attribute values would be revealed by their uncharacteristic surface features.

We define a set of string features that characterize a textual string's surface form. Each string feature describes a surface property of a string without concerning its literal meaning. Depending on the text language and the subject domain, string features can be defined at the character level or the word level. We will focus on the character-leveled features and their applications on Chinese IE. The same principles should be applicable to word-leveled features and other languages. We define a set of six string features, $SF = (S_c, S_p, S_s, S_e, S_l, S_n)$, that are used to evaluate each value v_i in an n-tuple. Given a set of n-tuple, $r = \{t_1, t_2, \ldots, t_m\}$, in the database initially loaded by an IE task, we will apply the set of string features to the set of n-tuple such that each value v_i in the database is evaluated by six string features. In other words, $SF(r) = (S_c\ (v_i), S_p(v_i), S_s(v_i), S_e(v_i), S_l(v_i), S_n(v_i))$, for each and every v_i in the database. The set of string features are specified as follows.

- String cardinality, denoted by S_c: the number of characters in the string.
- String k-prefix, denoted by S_p: the first k characters of a string, where k is a design parameter.
- String k-suffix, denoted by S_s: the last k characters of a string, where k is a design parameter.
- String entity, denoted by S_e: the full range of characters of a string.
- String lexicon, denoted by S_l: a true/false evaluation on whether the string matches with any of the known lexicon associated with a corresponding attribute.
- String numeral, denoted by S_n: a true/false evaluation on whether the string contains any numeral symbol.

3.2. Attribute Constraints

Assuming that domain knowledge of IE task subject is given, constraints on attribute values may be specified based on string features. Each string feature provides a potential classifying and screening criterion on data validity for an attribute. Recall that each attribute represents a concept of IE subject and attribute values are instantiations of the same concept in various symbolic forms. Whether the various valid symbolic forms of an attribute converge on certain string feature values depends on the nature of the corresponding concept. Some attributes may be associated with effective criteria on a single string feature, while others may need logical combinations of multiple string features to enable appropriate screening. Still other attributes may be difficult to find reliable string features at all. Nevertheless, string feature constraints, when available, provide a convenient and potentially effective way to discriminate invalid data.

The approach to examining string features of an attribute value against its known constrain for validity discrimination is both simple and straightforward. In addition, string feature constraints may be effective in detecting obvious errors due to IE's mistaken selection and insertion operations. However, the approach is not applicable to all attributes. It also requires human intervention in specifying known constraints. Finally, the approach may not be able to detect errors that are semantically incorrect but appear to have the right surface forms.

3.3. Majority Rule

Another approach is to adopt common string features of attributes based on group majority as classification standards. All the values of an attribute (a column of the database table) are considered as a population to be discriminated based on majority and minority. Each string feature provides some differentiating characteristics that divide the whole population into one larger group and one smaller group. Attribute values in the smaller group, as minority, are discriminated as invalid data based on majority rule. It is expected that some string features work better than others on certain attributes. The accuracy of detecting invalid data also depends on the boundary between majority and minority.

There are a number of strategies to draw the line between majority and minority. The most straightforward way is to calculate the occurrence percentage of each feature value for each string feature and rank them from the least to the most. Based on the ordered percentage distribution and a given threshold on accumulated percentages, a line can be drawn to separate minority from majority. For example, if a threshold of 20% is given, the subset of less significant groups with accumulated percentages less than 20% will be discriminated as minority. All other more significant groups are regarded as majority.

Overall, the approach to derive classification standards based on majority rule requires no prior knowledge on subject domains and allows fully automatic processing. However, the assumption here is the percentage of invalid data is significantly less than that of valid data. Otherwise, it would be difficult to identify anomaly based on the notion of uncharacteristic features as minority. From the standpoint of data cleansing, the assumption is reasonable since it may not be suitable, in the first place, to apply data cleansing to a database with very high percentages of incorrect data. Another potential problem is the performance variations of pairs of string feature and attribute. For each attribute, the best string feature for detecting invalid data may be different.

4. Experimental Evaluation

We apply the feature-based detection approach to a set of IE task results and evaluate its performance. The IE task domain is government personnel changes. The original documents are government gazettes publishing government personnel directives issued by the reigning President. Each government personnel directive authorizes government post appointment and dismissal of the named officials. The IE task is to transform government personnel directives into a set of structured information on the subject, such as person name, government unit name, position title, rank, type of changes, date, etc.

4.1. Performance Measures

With string features as the basis of error detection, we are essentially constructing binary classifiers that indicate the validity/anomaly of data. Each string feature can be used directly as a simple binary classifier. Or we can develop more enhanced classifiers by combining multiple string features. The standard performance measures for binary classifiers are summarized in a 2 x 2 confusion matrix, shown in Table 1.

Table 1: 2 x 2 confusion matrix

	Classified as Yes	Classified as No
Actual Yes Class	Number of True Positive (TP)	Number of False Negative (FN)
Actual No Class	Number of False Positive (FP)	Number of True Negative (TN)

From this matrix a number of standard metrics are specified to measure classification performance. Theses metrics are true positive rate, true negative rate, false positive rate, and false negative rate.

●*True positive rate*: TP-rate = TP/(TP+FN) is the percentage of positive cases correctly classified as belonging to the positive class;

●*True negative rate*: TN-rate = TN/(FP+TN) is the percentage of negative cases correctly classified as belonging to the negative class;

●*False positive rate*: FP-rate = FP/(FP+TN) = 1 – TN is the percentage of negative cases misclassified as belonging to the positive class;

●*False negative rate*: FN-rate = FN/(TP+FN) = 1 – TP is the percentage of positive cases misclassified as belonging to the negative class;

These four metrics measure the classification performance on the positive and negative classes independently, therefore, support objective evaluation even under skewed class distributions, as in our data set. The main objective of a classifier is to maximize the true positive and true negative rates. A perfect classifier will have true positive and true negative rates of 1. However, for most real world applications, there is always a tradeoff between TP-rate and TN-rate. The *ROC* (Receiver Operating Characteristic) graph can be used to characterize the tradeoff relationship between TP-rate and TN-rate of a classifier.

On a *ROC* graph, TP-rate is plotted on the Y-axis and FP-rate is plotted on the X-axis. The performance of a binary classifier, depicted by a (TP-rate, FP-rate) pair, corresponds to a point on the graph. The lower left point (0, 0) represents a classifier that never classifies a case as positive class. The opposite classifier, one that always classifies a case as positive class, is represented by the upper right point (1, 1). The upper left point (0, 1) represents a perfect classifier. The diagonal line x = y represents a classifier with a random guessing strategy. Any useful classifier must produce a point above the diagonal line. In general, a classifier is better than another if it produces a point closer to the northwest corner. For classifiers with parameters, different settings produce different *ROC* points. Plotting all the *ROC* points that can be produced by varying the parameters produces a *ROC* curve, representing the overall performance of the classifier.

4.2. Experimental Design

In order to evaluate our approach to feature-based detection for invalid textual data, we use a sampled subset of the initial database as our test data. The test data contains a total of 150,752 textual string values, with approximately 2.2% error rate. Our primary goal of the experiment is to assess the utility of each string feature and the overall performance of the approach in detecting invalid textual data from IE results. We observe that values of an attribute in IE results are symbolic form variations of a concept in the IE task subject. Our fundamental hypothesis is that certain string features are shared by string variations representing the same concept. These common string features may be found at the string's surface form level. As a result, effective detection of IE invalid values can be achieved by using the surface-form string features. The experiments are intended to verify the hypothesis and derive information on the overall performance of the approach.

We conducted two sets of experiments. In the first set of experiments, we apply subject domain knowledge to specify attribute constraints with selected string features. Classification performance measures are collected. The results provide information for us to analyze the applicability relationships between string feature types and attribute types. In the second set of experiments, it is assumed that no prior domain knowledge is available. Majority rule is applied for classification. We examine its performance and analyze the conditions of applicability. The combined results of the two sets of experiments provide empirical evidence for us to suggest the overall utility of string feature based detection for IE results.

4.3. Performance of Attribute Constraints

There are a total of 16 attributes in the domain of government personnel directives. Although the entire database was tested for anomaly detection, we only report performance measures on four representative attributes based on their distinguished content characteristics. In general, each attribute can be examined by any of the string features we defined. However, not all pairs provide useful anomaly detection. With subject domain knowledge, we can select testing pairs that are meaningful and avoid useless, even noisy, classification results. We summarize the selected combinations of attribute and string feature pair and their corresponding performance measures in Table 2.

Table 2: Performance metrics by attribute constraints

***ROC* point**	**attribute**	**string feature constraint**	**TP rate**	**FP rate**
A	person name	S_c (= 2 ~ 4)	**.9999**	**.8672**
B	person name	$S_p + S_l$ (first character $\in$ known)	**.9867**	**.1719**
C	person name	$S_c + S_p + S_l$	**.9926**	**.1016**
D	rank	S_n (= true)	**.6549**	**.2900**
E	rank	$S_n + S_l$ (= true)	**.6579**	**.0933**
F	issue number	S_n (= true)	**1.0**	**.3276**
G	cause of changes	S_l (= true)	**1.0**	**.2353**
H	government unit name	S_c (= 2 ~ 10)	**.1536**	**.2857**

Figure 1 shows the *ROC* graph of the selected testing pairs. The *ROC* points fall into three groups based on their locations. The first group is {A, B, C, F, G}, all of which have extremely high TP rates. This indicates that the string feature constraints specified for the particular attributes are highly successful in correctly classifying valid values. In addition, we can generally enhance their performance by adding more string feature constraints, as shown by the improvement from A to B to C. The second group includes D and E, which are only moderately effective in correctly classifying valid values. Note that the results were achieved by using binary string features only. Again, with more restricted validity constraints, more invalid values are detected correctly, as shown by the improvement from D to E. The third group contains H only, which is worse than random guessing classification. In fact, H is intended to show the possibility of a meaningless testing pair.

Based on these performance results, we make the following observations.

●Some attributes are strongly restricted by special types of value domains, which can be mostly represented by certain string features. Such a string feature is called the primary string feature of the corresponding attribute.

●For many attributes, there is a good chance of finding a primary string feature that provides satisfactory or even highly successful positive classification.

●The classification performance on an attribute can be enhanced by complementing its primary string feature with some secondary string features.

●Given subject domain knowledge, string feature based detection by attribute constraints may provide more than satisfactory performance.

The overall results seem to show that string features as attribute constraints are capable of effective data validity classification. We only show partial results in Table 2 to highlight some major points. However, we did confirm that by specifying more restrictive constraints with multiple string features *ROC* points of the attributes in our task domains all fall into area near the upper left corner. Finally, constraint specifications for attribute values involve only surface-form string features. Shallow domain knowledge is sufficed to provide the classification rules.

4.4. Performance of Majority Rule

The second set of experiments concerns with the use of string features in invalid value discrimination based on statistical majority. It is hypothesized that invalid values in IE results will be indicated by their unusual string feature values, which may be exhibited when comparing to other valid values of the same attribute. We conducted pair-wise testing on every combination of string feature and attribute. When classifying values of an attribute, frequencies of occurrence of string feature values are ordered from least significant to most significant. A threshold of accumulated percentage is used as the discrimination line between minority and majority. Based on majority rule, attribute values with minority string feature values are considered as invalid data. The parameter of accumulated percentage threshold is set at 1%, 3%, 5%, 10%, 15%, 20%, and 25%, respectively. This variation of parameter values produces a *ROC* curve, representing the overall behavior of the classifier.

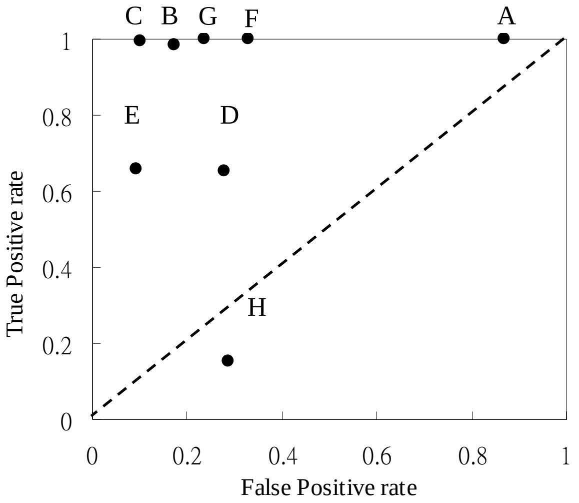

Figure 1: *ROC* graph of attribute constraints tests

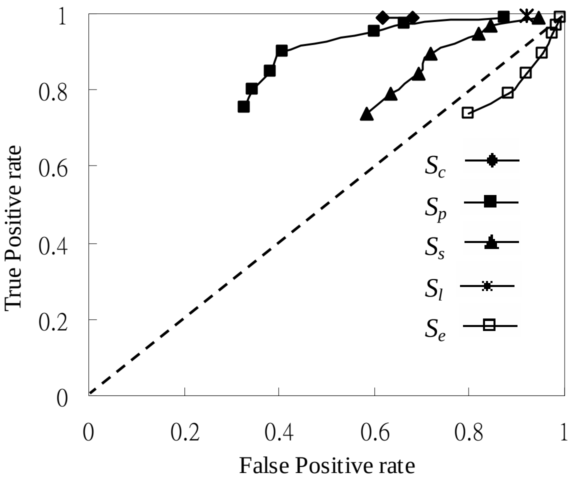

Figure 2: *ROC* graph of person name attribute

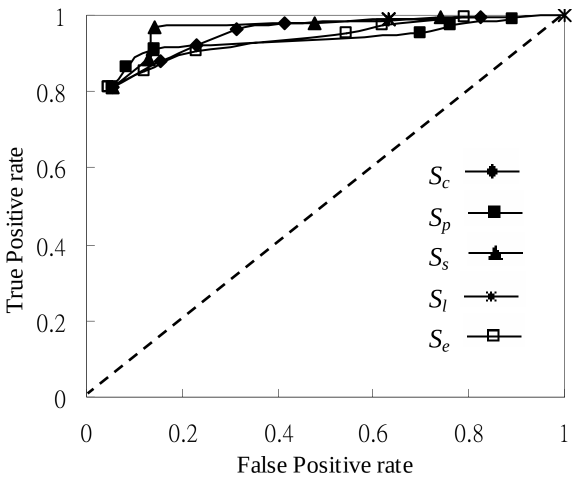

Figure 3: *ROC* graph of government unit name attribute

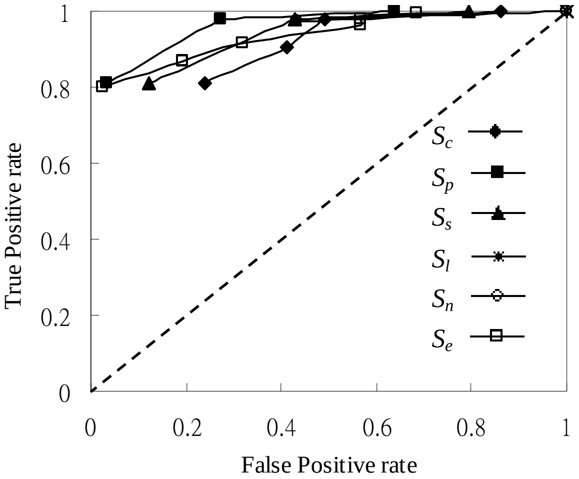

Figure 4: *ROC* graph of rank attribute

We report testing results of three representative attributes, as shown in Figures 2, 3, and 4. Accumulated percentage threshold of 1% corresponds to *ROC* points at the upper right corner, indicating the behavior of a highly positive classifier that considers at most 1% data as invalid. As the percentage threshold increases, the *ROC* points move to the left side with descending slopes. In

Figure 2, one string feature, S_e (string entity), produces a *ROC* curve worse than that of a random classifier. This is because frequency of occurrence of person name generally provides little indication on its validity. Many person names appear only once because they are involved in only one job change during the time covered in our data set.

Based on these performance results, we make the following observations.

- Some attributes, such as government unit name and rank, respond well to majority rule classification. Values of these attributes usually come from limited domains and are highly repeated. A single string feature is capable of providing effective discrimination with respectable results of 81% TP rate and 5% FP rate.
- Some attributes, such as person name, are more difficult to discriminate invalidity with majority rule classification. Values of these attributes are usually highly variable as a string unit. Classification performance by one single string feature is hardly satisfactory.
- Some invalid data are invisible from the angle of only one string feature and require cross examination from multiple perspectives. For example, invalid values of person name may also are strings of the same length with valid values of person name. However, these invalid strings of person name may be detected by examining their first character, which correspond to a person's last name. By combining multiple string features to provide finer grained discrimination, better performance can be expected.

The overall results show that when no prior domain knowledge is available majority rule discrimination is a feasible alternative for detecting invalid values. This suggestion is supported by the fact that attribute values in IE results usually exhibit high affinity with each other in certain aspect. The dominant form of valid values allows distinction with unusual variations by statistical majority.

5. Conclusions

IE techniques offer great potential for producing valuable databases from inventories of documents. However, before IE techniques can realize their full potential, the issue of data usability in IE results must be addressed. We propose a string feature-based approach to textual data profiling and invalid data detection. The approach is based on the observation that values of an attribute in IE results are symbolic form variations of a concept in the IE task subject and may exhibit a certain congruity with some string features. We conducted experiments to verify that effective detection of IE invalid values can be achieved by using the surface-form string features. The contribution of our work is to provide both analytical and empirical evidences for supporting the effective enhancement of IE results usability. The principles and the observations suggested in our paper are generic to IE task domains and languages and can be extended for further explorations in more sophisticated IE data cleansing.

6. References

[1] Agichtein , E., and Gravano, L. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries,* San Antonio, Texas, United States, June 02-07, 2000, pp. 85-94.

[2] Grishman, R. Information Extraction: Techniques and Challenges. In *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, M. T. Pazienza, (Ed.), Springer-Verlag, 1997.

[3] Muller, H., and Freytag, J. C. Problems, Methods, and Challenges in Comprehensive Data Cleansing. *Technical Report HUB-IB-164*, Humboldt University Berlin, 2003.

[4] Rahm, E., and Hong-Hai, D. Data Cleaning: Problems and Current Approaches. In *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4), December 2000.

[5] Riloff, E., and Lehnert, W. Information extraction as a basis for high-precision text classification. In *ACM Transactions on Information Systems*, 12(3), July 1994, pp. 296-333.