Application of Text Categorization to Astronomy Field

Huaizhong Kou*, Amedeo Napoli, and Yannick Toussaint

LORIA and INRIA-Lorraine
615, rue du jardin botanique, 54603 Villers-lès-Nancy, France
{huaizhong.kou,amedeo.napoli,yannick.toussaint}@loria.fr

Abstract. We introduce the application of text categorization techniques to the astronomy field to work out semantic ambiguities between table column's names. In the astronomy field, astronomers often assign different names to table columns at their will even if they are about the same attributes of sky objects. As a result, it produces a big problem for data analysis over different tables. To solve this problem, the standard vocabulary called "unified concept descriptors (UCD)" has been defined. The reported data about sky objects can be easily analyzed through assigning columns to the predefined UCDs. In this paper, the widely used Rocchio categorization algorithm is implemented to assign UCD. An algorithm is realized to extract domain-specific semantics for text indexing while the traditional cosine-based category score model is extended by combining domain knowledge. The experiments show that Rocchio algorithm together with the proposed category score model performs well.

1 Introduction

Text Categorization (TC) is the procedure of assigning one or multiple predefined domain-specific category labels to a free text document (category sometimes called "topic" or "theme"). Text categorization technologies have been widely employed to cope with various tasks that are based on the analysis of text content, such as categorical organization of news at Yahoo site.

In the astronomy research field, the volume of astronomy articles available in electronic forms grows increasingly over time and most of them contain one or more tables of observed data about sky objects, which consist of different columns. The tables contain observation data about various attributes of sky objects, such as temperature, luminary intensity, speed, position, rotation angle and so on. Also, they contain some data of astronomical instruments used to make observation. One column of a data table is about some attribute of sky objects. Actually, since there is not any standard about how to name table columns using a standard astronomy vocabulary, different astronomers often assign different names to the table columns of the same attributes of sky objects and consequently the semantics of data of table column are not clear. For example, there are 73 different column names in 3571 tables corresponding to the "Right Ascension"[1] attribute of observed sky objects. As a result, it is very hard to analyze and compare the data reported by different astronomers and many existing data mining technologies cannot be directly applied to discover astronomical knowledge. The ambiguity of table column names, which also covers semantics of column data, is one of the big problems for reusing and sharing of observed sky data.

---

* He is now with the Yellow River Conservancy Commission, China

[1] http://cdsweb.u-strasbg.fr/UCD/cgi-bin/ucd_stats?leaf=POS_EQ_RA_MAIN

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 32–43, 2005.

To solve such problems and ease both reusing and sharing of the observed data, one has to normalize the semantics of data of table columns reported by different astronomy researchers. One solution is that one semantically structured list of standard concepts is firstly constructed to represent the attributes of sky objects and then the columns are associated with the standard concepts. For this, one hierarchy of relatively standard concepts called Unified Content Descriptors (UCD) 1 has been already defined at Strasbourg astronomical Data Center (CDS) 1.

The UCDs can provide unified and unambiguous descriptions about the attributes of sky objects. The semantics of data of table column become explicit through establishing map relationship from table columns to the UCD concepts. For example, "POS_EQ_RA _MAIN" is an UCD, which represents the "Right Ascension" attribute of sky objects. When the 73 different column names mentioned above are assigned to it, the semantics of these 73 columns are clear. The UCD assignment is aimed to associate table's columns to the predefined UCDs with the goal that data about sky objects reported by different astronomers can be easily shared and analyzed. For this, one system of UCD assignment has been manually built and is used to assign UCD to table columns at CDS 1. As yet almost all of the 10^5 columns of the catalogues contained in VizieR have been associated with UCDs 1.

To support the UCD assignment, the *Readme* text files with specified format 6 are provided by astronomers when they upload their observation data documents. Among other things, the *Readme* files contain detail information about semantic of each column of table. Notice that there are already more than 10^5 columns associated with UCDs. These motivate us to test standard text categorization algorithms with the hope that UCD assignment system could be built automatically and the UCD assignment could be probably improved by text categorization technologies, which is based on analysis of the contents of column's explanation texts. To do this, we have adapted Rocchio algorithm.

The contributions of this paper include: the application of standard text categorization technologies to UCD assignment in the astronomy field is implemented, an algorithm of extracting domain-specific semantics is developed and a model of calculating category score, which can increase performance by 6.7% according to Rocchio algorithm, is also defined; the obtained results can provide support for the definition of UCD ontology that is ongoing parallel work in the frame of ACI-MDA project[2].

The rest of this paper is organized as follows. Section 2 is about UCD assignment and related information, then Rocchio algorithm is presented in Section 3 and a category score model is also proposed. Section 4 describes semantic enrichment for text index. The results and analysis of our experiments are reported in Section 5 while Section 6 concludes this paper by indicating some future work.

2 UCD Assignment

2.1 Objective of UCD Assignment

Figure 2.1 illustrates the scenario of UCD assignment. The left part shows that often there are observation data tables along with the articles published by astronomers. The

[2] http://cdsweb.u-strasbg.fr/MDA/mda_en.html

authors of articles name the columns of data tables at their will. Since there does not exist any standard in the astronomy field about naming the columns of observation data tables, different names are often given to the columns about the same attributes of sky objects and at the same time same names maybe are linked to different columns about different attributes of sky objects. For example, there are 73 different column names for the “right ascension” attribute, including “RA”, ”Rao”, ”RAL”, ”RAG”, ”RA1984”, ”RAX”, ”RAK”, and so on. Such as, automatic analysis over different data tables is almost impossible.

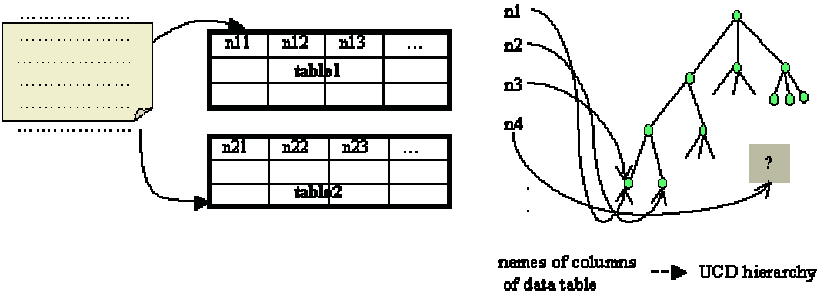

Fig. 2.1. Data table and UCDs.

The proposal of UCD by CDS is a great step toward standardizing the names of table columns and easing automatic data analysis. All UCDs together constitute one hierarchy tree of standard concepts in the astronomy field for naming table columns, like shown by the right part of Figure 2.1. The objective of UCD assignment is to match columns of data tables to a concept node in the UCD tree. The UCD assignment can be performed by analyzing the *ReadMe* file.

On the other hand, from the point of view of data integration, if we take the structures of data tables independently defined by different astronomers as local schemas of observation data and UCD hierarchy as global schemas accordingly, the problem of assignment of UCDs to table columns is just like one of mapping local data schema to one global schema, which is essential step toward transparent use of data. Once local schemas are mapped to the global schema, many analysis across data sources can be performed.

In our case, schema mapping is aimed to establish correspondences for table’s columns to UCDs in the UCD hierarchy. In the practice, description matching is one of approaches to schema mapping. By description matching, comment texts in natural language about the semantics of schema elements will be linguistically evaluated to map the elements of local schemas into the global schema. In the context of UCD assignment, the *ReadMe* files provide such comment texts about the semantics of table structures, see Section 2.3.

2.2 Schema of UCD

The UCD schema is a 4-level hierarchy tree that is firstly presented in 5. It contains 1417 nodes, including 1380 leaf nodes and 37 non-leaf nodes. Actually only 1183 UCDs are used in *VizieR*.

UCD is defined by the pairs (name, definition), where the name is identification of UCD used in *VizieR* and the definition defines semantic meaning of UCD. For example, the followings are three UCDs[3]:

- AT_COLL Atomic Collisional Quantities
- AT_COLL_EXCIT-RATE Collisional Excitation Rate
- AT_COLL_STRENGTH Collisional strength

AT_COLL is an internal node for atomic collision quantities while the second and the third UCD are for two different quantities of atomic collision, excitation rate and strength respectively.

UCD hierarchy tree provides an unified schema of concepts at the global level. On the top of such standard schema various correlation analysis of observation data parameters supplied by different astronomers can be performed. See 1 for the entire UCD tree.

2.3 ReadMe

Given a catalogue, its *ReadMe* 6 is a text file that contains all necessary information to interpreter and locate the contents of catalogues. It is composed of many sections, such as abstract, keywords, notes and etc.. Among them, the section *Description of data table* is most important for assigning UCDs.

The section *Description of data table* consists of many rows, each of which describes the semantics of one data table column in five fields: bytes, format, unit, label and explanation. The functions of the five fields are as follows:

- Bytes: the position of column data.
- Format: data format of column content.
- Unit: used for the data of column content.
- Label: column name.
- Explanation: a short text to explain the semantic information of column content.

We cite some rows of this section[4] as follows:

	Bytes	format	unit	label	explanation
•	31-32	I2	arcmin	DEm	Declination J2000 (minutes)
•	1- 12	A12	---	MACS	Designation
•	14- 15	I2	h	RAh	Right Ascension J2000 , Epoch 1989.0 (hours)
•	17- 18	I2	min	RAm	Right Ascension J2000 (minutes)
•	20- 25	F6.3	s	RAs	Right Ascension J2000 (seconds)
•	27	A1	---	DE-	Declination J2000 (sign)

Take the first row as an example. “31-32” is the position of column’s data in the data table file; “I2” is format of column’s data and it means two integers; “arcmin” is the unit that is associated with the table column named as “DEm”; the explanation “Declination J2000(minutes)” describes the semantic of the columns. Three fields: unit, label and explanation are actually used to assign UCDs to table columns 5. See 6 for detail of *ReadMe* file.

[3] http://cdsweb.u-strasbg.fr/viz-bin/UCDs

[4] http://vizier.u-strasbg.fr/doc/catstd-3.1.htx

2.4 Frame of UCD Assignment

Figure 2.2 shows the frame of UCD assignment. At large, it consists of two parts. The first one that is in dotted-line rectangle is aimed to learn the text classifier; the second one is to assign table's columns to UCDs. The corpus for learning is made up of example columns with their explanations contained in the corresponding *ReadMe* files and the UCDs that are already assigned to the example columns in the VizieR system at CDS. Learning the text classifier is a supervised process, which includes selection of vocabularies, text index, estimation of text classifier's parameters and etc..

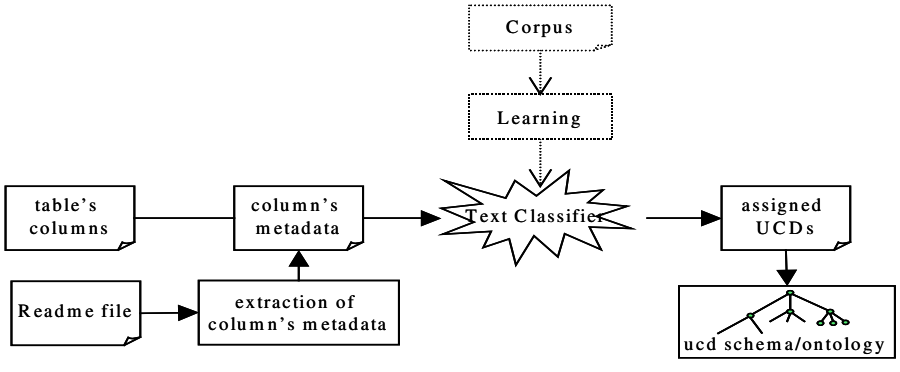

Fig. 2.2. Frame of UCD assignment.

Once the text classifier is built, it can be used to assign table's columns to UCDs. To assign table's columns to UCDs, the column's metadata are firstly extracted from the corresponding *ReadMe* file. Then they are fed to the learned text classifier. The text classifier will return one or more UCDs to the table's columns on analyzing the text contents of the metadata.

2.5 Related Work

At CDS, one system of UCD assignment has been manually built mainly using units, column's labels and explanations. As for any manually building system there is a very common agreement. That is, it is time consuming, heavily dependent of limit of knowledge of domain experts and it is very difficult to update the system over time. The CDS's system is certainly not an exception.

More than 10^5 columns have been assigned to UCDs by CDS's system until now. For a given table column described by the metadata section of *ReadMe* file, UCD assignment is performed in two steps: firstly several most possible UCDs are returned by CDS's system on calculating relevance score for every UCD; then astronomers can select one UCD from the returned UCDs to assign it to the column. If neither of returned UCDs is suitable, astronomers have to choose one UCD from the UCD schema by themselves for the column. For this, sometimes astronomers have to study the content of article to finally decide one UCD. See 1 for more information about how to use CDS's assignment system.

3 Text Categorization Algorithm

3.1 Rocchio Algorithm

Rocchio algorithm is the relevance feedback-based optimal query creation algorithm, which has been successfully used in IR as of its early days. It has been adapted to categorize text 3. Being applied to document categorization, Rocchio firstly builds one "centroid vector", sometimes called "conceptual vector", for each category c_i (i=1, 2,..., m) averaging the vectors of documents assigned to the category with the formula (3.2).

$$\vec{c}_i = \frac{1}{|c_i|}\sum_{d\in c_i}\vec{d} \tag{3.2}$$

$$simil(c_i, d) = \cos(\vec{c}_i, \vec{d}) \tag{3.3}$$

To categorize a given document d, the similarities between the vector of document d and the centroid vectors of all category c_i (i=1, 2,..., m) are calculated using cosine function-based similarity model (3.3). Such similarities are used as category scores. Then all categories are ranked in the decreasing order of category scores, and so a ranked list of categories is obtained. Finally, some strategy is taken to decide the categories to which the document d is assigned. Rocchio algorithm assumes that the centroid vector of category can present the concept model of category. It performances well if the centroid vectors can characterize the concepts contained in categories.

3.2 Model of Calculating Category Score

In our implementation of UCD assignment system, three factors are combined to calculate category score in the following formula (3.4).

$$\begin{aligned} score\ (\ column, UCD\ C) &= unitRevelance(unit, unitList, \beta) \times \\ &(\ \alpha \times Simil(vector\ of\ C\ definition, column's\ explanation\ vector) + \\ &(\ 1-\alpha) \times ScoreByAlgo(C, column's\ explanation\ vector)) \end{aligned} \tag{3.4}$$

$$unitRelevance(unit, unitList, \beta) = \begin{cases} 1, & if\ unit\ is\ in\ unitList \\ \beta, & if\ unit\ is\ not\ in\ unitList \end{cases}$$

where $\alpha \in [0,1]$ and $\beta \in [0,1]$, *ScoreByAlgo*(. , .) is the category score calculated by (3.3), *Simil*(vector of *C* definition, column's explanation vector) is calculated using cosine function and *unitRelevance*(.,.) measures the impact of units on relevance of UCD and column. The model in (3.4) is equal to *ScoreByAlgo*(. , .) if α is set to 0 and β to 1.

In the case of UCD assignment, some UCDs are very similar and the explanation texts of columns belonging to them share very much terms. On the other hand, the definition texts of UCDs often contain significant terms for assigning UCD. Due to this fact, we introduce the similarity between UCD's definition and column's explanation text by *Simil*(vector of *C* definition, column's explanation vector). We hope that it can help to identify similar UCDs.

The introduction of the function *unitRelevance*(.,.) is based on the assumption that one UCD should be punished by reducing its category score if the list of accepted units of the UCD does not contain the unit of the column to be categorized. In other words, it takes 1 as its value if column's unit is in the list of accepted units of the UCD and $\beta \in [0,1[$ as its value otherwise. Here we assume that a relatively complete list of units used by table columns for a given UCD can be obtained if the corpus covers many enough cases of different columns. Setting β as 0 means that a column is certainly not assigned to an UCD if the unit of the column is not in the list of accepted units of the UCD while β with value as 1 means that units are not taken into account in calculating category score.

4 Semantic Enrichment for Index

4.1 Simple Words and Semantic Enrichment

Text index mainly consists of first identifying index words from texts of documents and of then measuring the importance of index word to represent the content of document. In the practice of traditional IR and text categorization, simple word-based text index is often used and documents are taken as a bag of words without distinguishing semantic information of words. Intuitively simple word-based index technologies have some limitations 7, including such as synonym, polysemy, local context and so on.

Assumed that more semantic information is used and higher precision performance can be achieved, some works on semantic enrichment-based index have been reported by different researchers 72. For example, context-specific phrases are extracted using information extraction technologies in 7, furthermore they are applied to text index and high precision of text categorization is achieved. Domain-specific concepts are extracted using probabilistic latent semantic analysis 2 and these domain-specific concepts are used to supplement simple index words. Their experimental results have shown that text categorization performance has been improved.

4.2 Astronomy Domain-Specific Semantic Enrichment

We also strongly believe that domain-specific semantic information can improve text categorization performance. It seems that semantic enrichment is more important in the case of UCD assignment mainly because the document texts are very short (only about 4 words) and information contained in document text is relatively poor. And often simple word-based index cannot discriminate certain close topics, for which semantic enrichment is indeed essential. For example, the following four UCDs are very similar:

- PHOT_FLUX_IR_12 : Flux density (IRAS) at 12 microns, or around 12 microns (ISO at 14.3)
- PHOT_FLUX_IR_25 : Flux density (IRAS) at 25 microns
- PHOT_FLUX_IR_60 : Flux density (IRAS) at 60 microns
- PHOT_FLUX_IR_100 : Flux density (IRAS) at 100 microns

Table 4.1. Example of UCD and texts of column's explanations.

UCD	Text of column's explanations
PHOT_FLUX_IR_12	Flux density at 12 micron
PHOT_FLUX_IR_12	[0,] IRAS flux density at 12micron
PHOT_FLUX_IR_12	Estimated IRAS 12 micron flux
PHOT_FLUX_IR_25	IRAS flux at 25 micron
PHOT_FLUX_IR_25	25 micron IRAS flux
PHOT_FLUX_IR_100	Flux density at 100 micron
PHOT_FLUX_IR_100	IRAS flux density at 100micron

```
1. Input : doc_tokens
2. String  regX1 = "[0-9]*[\\.]?[0-9]*\\s*[\\-]?\\s*[0-
   9]*[\\.]?[0-9]+";
3. String  regX2 = "[0-9]*[\\.]?[0-9]*\\s*[\\-]?\\s*[0-
   9]*[\\.]?[0-9]+\\s*[a-zA-Z]*";
4. List indexTerms;
5. while (hasMoreTokens){
6.    String word  =  nextToken();
7.    indexTerms.add(word);
8.    if (matches(regX1, word)){
9.       String unit = nextToken();
10.      if (isValidUnit(unit)){
11.          indexTerms.add(word+unit);
12.      }
13.     indexTerms.add(unit);
14.   } else if (matches(regX2,word)){
15.     String  unit  =  extractUnit(word);
16.     if (unit!=null)){
17.       indexTerms.add(unit);
18.     }
19.  }
20.}
21.return indexTerms;
```

Fig. 4.1. Domain-specific information extraction algorithm.

The column's explanation texts labeled with them often share much of common simple words like shown by Table 4.1. If here the number "12", "25" and "100" are individually treated, they are just same as mathematic number as themselves. As a result, the local context semantic information associated with them is certainly lost and this may mislead UCD assignment.

One of solutions to such problems is to combine these numbers with special words immediately following them (such as "micron" in our example) and enrich index semantic information. We will use, for example, "12", "micron" and "12micron" instead of only "12" and "micron" as index terms to represent the explanation "Flux density at 12 micron".

In our system, we extract two types of piece of information to enrich index semantic information at present:

a) Domain-specific patterns will be identified from explanation's texts. For example, "12 micron" will be extracted from the text "Flux density at 12 micron" and "2-10kev" from "The 2-10 keV count rate (2)".

b) Domain-specific simple words will be identified from some composite words. For example, “micron” will be identified from the text “[0,] IRAS flux density at 12micron”. This allows represent the relationship between “micron” and the text using both “micron” and “12micron” instead of only “12micron”.

Our information extraction algorithm in Figure 4.1 actually is knowledge-based because an astronomy domain-specific unit dictionary is used to guide extraction of both domain-specific patterns and domain-specific simple words. During the preprocessing of text, the extraction algorithm will be activated each time that predefined trigger pattern is encountered in order to identify index words from texts of documents. The trigger patterns are defined using regular expressions for every type of piece of information to be extracted.

The input *doc_tokens* is tokenized document text with space as delimiter, regX1 and regX2 are two trigger patterns respectively corresponding to the types a) and b). If the current *word* matches regX1 (line 8), the method *isValidUnit*() will be triggered to check if the following word called *unit* is valid unit (line 9-11); if it matches regX2 (line 14), the method *extractUnit*() will be triggered to try to extract possible *unit* contained in *word* (line 15-18). Both *isValidUnit*() and *extractUnit*() are based on the astronomy unit dictionary. The extracted words will be added into index term list to supplement simple words rather than substitute them.

5 Experiment Results and Discussions

5.1 Design of Experiment

We notice that the numbers of columns assigned to each UCD are very different. For example, 2481 columns are assigned to the UCD “ ERROR” but only 1 column is assigned to many UCDs. Among 874 UCDs, some moderate UCDs whose frequencies are between 30 and 100 inclusive are selected and the columns that are assigned to these moderate UCDs are picked up to make up the corpus. Finally, our corpus includes 93 UCDs and 4904 columns: 3371 for training and 1533 for test. Two approaches to text indexing are implemented. The first approach to index is trivial, shown as follows:

- Text documents are tokenized into individual words.
- 318 English stop words are removed, such as “the”, “of”, “on” and so on.
- Non-alphabetic characters are removed with the except that “-” and “_” are kept.
- Variants of words are transformed to their dictionary original form with the help of WordNet 2.0[5]. For example, navigating, navigated =>navigate; thought, thinking =>think.
- Total 3228 words are obtained and all of them are used to index text documents.

In addition to the first 4 above steps, the second index approach extracts some domain-specific information using the algorithms presented in the Section 0 to enrich semantic information. By the second approach, the total 3636 words are obtained and used to represent text documents.

The number of words used for text index is very lower than mostly reported cases in the literature. The average number of words of documents is only 4. RCut thresh-

[5] http://www.cogsci.princeton.edu/cgi-bin/webwn2.0

olding strategy 8 is used to decide which category/ies is/are assigned to text documents. To evaluate the performance, micro- and macro- average recall, precision and F1 measures 4 are used. We take column's label plus explanation as documents in our experiments. That is, document= "column's label" + "column's explanation".

5.2 Results

Table 5.1 shows the experiment results by Rocchio for RCut = 1 and 3. For each case of RCut, the results are divided into 3 rows: the first row is the results by the first approach to text index without semantic enrichment treatment and the normal score calculated by Rocchio, where $\alpha = 0$ and $\beta = 1$; the second row is the results by the second approach to text index with semantic enrichment treatment but not taking into account the impact of units on categorization, where $\alpha = 0.3$ and $\beta = 1$; the third row is the results by the second approach to text index with semantic enrichment treatment and taking into account the impact of units on categorization, where $\alpha = 0.3$ and $\beta = 0$. $\beta = 0$ means that the categories will be rejected if the lists of units accepted by them do not contain unit associated with the columns to be categorized. We found that the best results are reached by Rocchio if α=0.3 and β=0.

Table 5.1. Performance by Rocchio for RCut =1 and 3.

RCut	α	β	micro-average			macro-average		
			r	p	f1	r	p	f1
1	0	1	77.3	77.1	77.2	78.2	79.6	76.7
	0.3	1	78.4	78.1	78.2	79.1	80.1	77.2
	0.3	0	81.3	82.1	81.7	81.7	83.7	80.4
3	0	1	90.8	30.6	45.8	91.1	36.6	50.1
	0.3	1	92.1	30.8	46.2	92.1	37.1	50.5
	0.3	0	93.2	35.9	51.8	93.1	44.1	56.8

5.3 Discussion

The results in Table 5.1 show that Rocchio together with our category score model performs well. The main idea of Rocchio is to build a centroid vector for each category and then the centroid vectors are explored to represent the main concepts discussed by the categories. If such centroid vectors indeed characterize the concept patterns contained in the categories, the Rocchio algorithms can work well.

In the case of UCD assignment, the centroid vectors for the UCDs can really represent main concept patterns of UCDs. In fact, the top 20 representative terms of centroid vectors can often describe the main concepts discussed by UCDs. For example, the UCD "PHOT_FLUX_IR_100" is about " Flux density (IRAS)" at "100 microns"

and the terms related to it are “flux”, “density”, “iras” and “100microns”. They are all present in the top 20 terms of its centroid vector.

Let us have a look at the contribution of semantic enrichment and column’s units to UCD assignments. According to Table 5.1, the micro-average performances of UCD assignment by Rocchio algorithm with semantic enrichment index increases by 1% compared to one by the trivial text index approach. This is due to the fact that semantic enrichment can improve text index. The following example can also confirm this fact.

Example. For the text “F100um Flux density at 100 micron”, its document vector is “100micron 0.5897, f100um 0.5897, micron 0.4172, density 0.2837, flux 0.2234” if semantic enrichment is performed; otherwise “f100um 0.7135, micron 0.5478, density 0.3432, flux 0.2703”. Its correct UCD is “PHOT_ FLUX_IR_100”. In these two cases, the possible 3 top UCDs returned by our system are as follow:

- PHOT_FLUX_IR_100 0.4690 0.3775
- PHOT_FLUX_IR_25 0.2551 0.3883
- PHOT_FLUX_IR_12 0.2512 0.3864

Here, the first and second numbers are the category scores calculated with semantic enrichment and without semantic enrichment respectively. Obviously, the result by semantic enrichment index is more reasonable.

We also notice that the gain of performance with semantic enrichment index together with column’s units is of at least 4% in terms of micro-average measures. This implies that units of columns can play important rule in correctly assigning UCD. This also provides useful hints for constructing UCD ontology. That is, information about units must be closely studied when defining UCD ontology.

The following example shows how column’s units improve the performances of UCD assignment.

Example. For the column whose label is “(O-C)Rho” and its explanation is “(O-C)Rho (O-C) in separation, arcseconds” and the linked unit is “deg”. Its correct UCD is “FIT_RESIDUAL”. The followings are the top 6 UCDs returned by Rocchio that are in descending order of category scores:

- CLASS_STAR/GALAXY 0.0767
- POS_EQ_DEC_OFF 0.0734
- POS_EQ_RA_OFF 0.0648
- FIT_RESIDUAL 0.0510
- PHYS_DISTANCE_TRUE 0.0162
- PHOT_COLOR_EXCESS 0.0053

The list of units accepted by the UCD “CLASS_STAR/GALAXY” does not contain the unit “deg”, so “CLASS_STAR/GALAXY” is rejected if β =0. In fact, the units accepted by “CLASS_STAR/GALAXY” is only “%”. In this case, the final top 3 UCDs returned by categorization system include “POS_EQ_DEC_OFF”, “POS_EQ_RA_OFF” and “FIT_RESIDUAL”, whose lists of units contain the unit “deg”. The correct UCD “FIT_RESIDUAL” is returned if RCut=3 and β=0, while the first 3 top UCDs would be returned and “FIT_RESIDUAL” not returned if β =1. It means that the number of correct UCDs returned can be increased if rejecting the UCDs with the list of accepted units that does not contain the units of unknown col-

umns. This example also confirms our assumption made in the Section 3.2. That is, one UCD should be punished by reducing its category score if the list of accepted units of the UCD does not contain the unit of the column to be categorized.

For RCut =3, both micro- and macro-recall are higher than 90% while the precisions are very low, since 3 possible UCDs are returned. Actually, the probability that the returned 3 possible UCDs include one correct UCD is 95%. Compared to the UCD assignment system at CDS, the results show that our UCD assignment system reaches very good performance. In addition, our model of category score defined by (3.4) increases the performance by 4.5% in terms of micro-average F1 and 6.7% in terms of macro-average F1 for RCut =1 and 3 respectively.

6 Conclusion and Future Works

The ambiguities of data table column's names reported in astronomy articles is a big problem for sharing observation data about sky objects in the astronomy field. By introducing the application of text categorization techniques, we attempt to cope with this problem and automatically build up an UCD assignment system. The experiment results have shown that domain-specific semantic enriching and utilization of domain-specific knowledge can improve performance of categorization.

In the framework of ACI-MDA, the parallel work focused on constructing an ontology of UCD is ongoing. In the future work, we will collaborate the knowledge of ontology of UCD and the built text categorization system.

Acknowledgements

The authors would like to thank our astronomy colleagues of CDS. The research is supported by ACI scientific research funds of France.

Reference

1. CDS, http://cdsweb.u-strasbg.fr/; VizieR, http://cdsweb.u-strasbg.fr/viz-bin/VizieR; UCD assignment, http://cdsweb.u-strasbg.fr/UCD/assign/
2. L. Cai and T. Hofmann, Text categorization by boosting automatically extracted concepts, Proceedings of the 26th SIGIR conference, Canada, pp.182-189, 2003.
3. T. Joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, In the 14th Int. Conf. Machine Learning, pp.143-151,1997.
4. H. KOU, Intelligent Web Wrapper Generation Using Text Mining Techniques, PhD thesis, University of Versailles, July 2003.
5. Ortiz P. F., Ochsenbein F., Wicenec A., & Albrecht M., ESO/CDS Data-mining Tool Development Project, in ASP Conf. Ser., Vol. 172, eds. D. M. Mehringer, R. L. Plante, & D. A. Roberts (San Francisco: ASP), 1999.
6. Readme: http://vizier.u-strasbg. fr/doc/catstd.htx
7. E. Riloff, W. Lehnert, Information extraction as a basis for high-precision text classification, ACM Transactions on Information Systems, Vol.12, Issue 3, pp. 296 – 333, 1994.
8. Y. Yang, A study on thresholding strategies for text categorization, Proceedings of the 24th ACM SIGIR Conference, pp.137-145, 2001