TDU Systems for MuST: Attribute Name Extraction, Text-Based Stock Price Analysis, and Automatic Graph Generation

Minoru Yoshida† mino@r.dl.itc.u-tokyo.ac.jp
Takahiro Sugiura‡ sugiura311@gmail.com
Takamasa Hirokawa‡ hirokawa@cdl.im.dendai.ac.jp
Kouichi Yamada‡ yamada@im.dendai.ac.jp
Hidetaka Masuda‡ masuda@im.dendai.ac.jp
Hiroshi Nakagawa† n3@dl.itc.u-tokyo.ac.jp

†University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033
‡Tokyo Denki University
2-2, Kanda-Nishiki-cho, Chiyoda-ku, Tokyo 101-8457

Abstract

In this paper, we report our participation in MuST as the Tokyo Denki University team. We participated in free tasks with two systems: a system for attribute-name extraction and a system for analysis of relations between texts and stock prices. We also participated in the T2N task with a system for generating graphs from news articles automatically. We describe the algorithms of these three systems and discuss about the results of the T2N task.

1 Introduction

The Tokyo Denki University (TDU) team participated in MuST with two systems for free tasks and one system for the T2N task. This report gives a description of these three systems and discussion of the results for the T2N task.

In section 2, we report our system that extracts attribute names from news articles. Section 3 describes our research on extracting relations between words and stock prices. Section 4 reports our participation in the MuST T2N subtask with the system based on the one described in Section 2. In section 5 we summarize this paper.

2 System-1: A System for Extracting Numerical Values and Their Attribute Names

In this research, we propose a system for automatic extraction of attribute names[1]. Our system uses Support Vector Machines[11] trained on the MuST corpus to extract attribute names related to numbers. We report the performance of the system on attribute extraction from the MuST corpus and show that our SVM-based system outperformed a simple baseline method that uses dependency relations extracted by dependency parser CaboCha[3]. We also show the results of a *cross-topic experiments* that have not been reported in other previous studies.

2.1 MuST Corpus

Figure 1 shows an example text from the MuST corpus. The MuST corpus consists of news articles by Mainichi Newspapers with tags like `<name>`(*i.e.*, attribute names)[2], `<val>`(*i.e.*, numerical values), `<rel>`(*i.e.*, relative values), etc. Texts in the corpus are divided into topics like *Gasoline*, *Nikkei Stock Average*, etc.

2.2 Attribute Name Extraction

We obtained the following insight into attribute names based on reviewing on the MuST corpus and preliminary experiments.

[1] In this paper, we use the term *attribute name* as a synonym of *name of statistic*.

[2] We use all the strings with `<name>` tags as attribute names regardless of which options the `<name>` tags have.

```
<unit stat="メーカー毎の PC 出荷シェア">
  <par> NEC など昨年の上位 5 社</par> の
  <name> シェア</name> は
  <pro ref="前年比" id="9801220"> 同</pro>
  <rel type="prop">3.1 ポイント</rel> 低い
  <val>82.7%</val>
  となった
</unit>.
```

Figure 1. Example: MuST Corpus

- Attribute names are likely to appear just before their values.
- Attribute names are likely to have dependency relations with their values.
- Attribute names are likely to depend on the same verb as their values.
- Attribute names are likely to be noun compounds (*e.g.,* 内閣支持率).

The first observation is easily used as extraction rules or features for machine learning. The last three observations led to inclusion of the following features in the feature set used in SVMs.

- Dependency relations extracted by the dependency parser CaboCha.
- Noun compounds extracted by the term extraction tool Gensen.

2.3 Extracting Attribute Names with Support Vector Machines

Support Vector Machines are machine learning algorithms for binary classification. Given a collection of labeled examples $(x_1, l_1), \cdots, (x_m, l_m)$ where x_i is a vector and $l_i \in \{+, -\}$, SVMs maximize margins between positive examples and negative examples. Each x_i is a vector for each example (which is, in our setting, each character in texts) in which a value of each dimension is determined as its corresponding feature value.

The system performs morphological analysis by Chasen[1], dependency analysis by Cabocha, and noun compound extraction by Gensen[5] on given texts.

Table 1 shows a list of features used in our model. The feature set includes basic features like word types and POS[3] as well as some advanced features like frequencies with numbers and dependencies on numbers as mentioned in the previous section. Character-position features are based on the Start-End method[10]. In this method, the first character in each morphology is tagged with `B`, the last character is tagged with `E`, inner characters are tagged with `I`, and one-character morphology is tagged with `S`. The "dependency relation" feature has four values: "the number depends on the target bunsetsu", "the target bunsetsu depends on the number", "depends on the same bunsetsu as the number", and "others". The "noun compounds" feature indicates the target word is contained noun compounds extracted by Gensen.

[3]These features are determined based on the work Nakano et al.[6]

Table 2. 7 types of characters.

Hiragana, Katakana, Digits, Alphabet (lower letter), Alphabet (upper letter), Spaces, Others.

Table 3. Topics Used in Our Evaluation

Gasoline, Sony, PC, Beer, CommunicationDevice, TrafficAccident, Population, Estate, LandPrice, MajorLeague, UnemploymentRate, PolicicsTrend, NikkeiStockAverage, Movie, BusinessForecast, Car, Olympic, AirConditioner, ExchangeRate, FamilyBudget, Sales, ElectronicCompany, SuperMarkets, CommodityPrice

The attribute name extraction is performed per character by labeling each character with B/I/O (Beginning/Inner/Other) tags popular in natural language processing like named-entity extraction. In labeling each character, the system looks at two characters adjacent to the left of the target character and also two characters adjacent to the right. Features are extracted for these five characters (including the target character itself) and used in SVM training/test. We adopt a leftward processing method in our name extraction. In this method, the last character in each sentence is labeled first, followed by labeling the second-last character, and so on. It is based on our intuition that suffixes are important clues to find attribute names. We use additional features that indicate if the next two characters are tagged with B or I. These additional features contribute to improving the consistency of name extraction (*i.e.,* continuous characters tend to be extracted as one attribute name).

2.4 Evaluation

Our attribute name extraction system is evaluated on the task of extraction of words tagged with `<name>` tag in the MuST corpus. We used 518 articles out of 581 from the MuST corpus. The topics used in our evaluation are listed in Table 3.

We conducted two types of experiments. One is the 5-fold cross validation on randomly-partitioned 5 document sets, and the other is *the cross-topic validation* in which the attribute names are extracted from documents in one topic by the SVMs trained on documents

Table 1. A List of Features Used in Our Model.

Feature	Description
Character type	7 types shown in Table 2.
Character	Character itself.
Character position in words	4 types (Begin, End, In, Single)
Word	Word itself.
Part Of Speech	127 types in Chasen outputs.
Frequency with numbers	The number of appearances just before numbers
Dependency relation	4 types for relation to numbers
Frequency of numbers/verbs	A pair of frequencies of related verb/number
Noun compounds	Noun compounds extracted by Gensen.

Table 4. Evaluation Results

Method	F-measure	Recall	Precision
Baseline	0.38	0.42	0.34
Cross-topic	0.65	0.53	0.82
5-fold	0.82	0.78	0.87

in *all other topics*. The latter evaluation gives us some insight about how well the trained SVMs will perform on documents from unseen topics. For example, when testing on the topic *Gasoline*, we can see how well the SVMs trained on the topics other than *Gasoline* extract attribute names from documents in topic *Gasoline*.

We computed precision and recall values for each test, followed by the calculation of F-measure (harmonic mean) and taking averages of precision, recall, and F1 values over all tests. Recall and precision are defined as

$$Recall = \frac{\text{\# of correct extractions}}{\text{\# of correct answers}}$$

$$Precision = \frac{\text{\# of correct extractions}}{\text{\# of extractions}}$$

Note that "correct extractions" includes the partial extractions like the case that "失業者数" is extracted instead of the correct answer "完全失業者数". We also evaluated the performance of the system[9] that uses simple extraction rules on dependency relations extracted by CaboCha.

2.5 Results

Table 4 shows evaluation results by f-measure, recall, and precision. We observed that our SVM-based extraction system outperformed the rule-based baseline system. The result of 5-fold cross validation was much better than that of the cross-topic validation. This result suggests that topic-overlaps between training and test sets greatly improve the extraction performance.

The result of cross-topic validation was better than the baseline method (although it was worse than the 5-fold cross validation). This results indicate that our SVM-based system was more tolerant to unseen topics than hand-crafted general rules. Precision was improved for almost all topics and recall was especially improved for some topics including "Population", "Estate", and "Movies" from the baseline.

However, there were also some exceptions where recall or precision was decreased. These exceptions include, for example, the topic "ExchangeRate" for which recall was 0. Articles in this topic have some particular kind of expressions like "1 dollar = 132 yen" for which our SVM features like dependency relations and noun compounds did not work well. Precision was low for the topics "Car" and "ExchangeRate". Errors for the topic "Car" included, in addition to normal errors, the annotation disagreement between the system and corpus (*e.g.*, words like *export*, *sales* extracted by our system were not tagged as attribute names in the MuST corpus).

2.6 Applications

We implemented some application systems based on our attribute-name extraction algorithm.

One system is the *number search engine*, which provides a function to rank documents by the values of some specific attributes. It provides a new way to rank documents. For example, by ranking documents by the amount of sales, we can find articles about active companies effectively.

Another system is the *number plot interface*, which collects values of the input attribute from given articles, and plots them on the graph. We can survey the general tendency of values at a glance by using this system.

We also implemented the *graph generation system* to participate in the MuST T2N task. The detailed description of this system is given in section 4.

2.7 Conclusions and Future Work

This section described our system to extract attribute-names from news articles. Our system uses

SVMs with some special features like dependency relations with numbers or inclusion in noun compounds. We evaluated the system performance on the ordinary 5-fold cross-validation as well as the cross-topic validation which is important to see how much the system is tolerant to unseen topics. The results showed that the system with SVMs was more tolerant than the system with hand-coded extraction rules, although the result of cross-topic validation gave a worse f-measure values than the 5-fold cross validation. Future work includes development of more intelligent mining system based on extracted values.

3 System-2: A System for Analysis of Relations between News Articles and Stock Prices

In this section, we report our analysis of relations between words and stock prices as the first step towards leveraging of *words in texts* for stock price prediction. Our approach is to extract frequent words in news articles both on the days with increasing stock prices and on the days with decreasing stock prices. Comparing these two word sets suggest some clues that will be helpful for predicting future stock prices.

Current techniques to forecast stock prices typically make a forecast by mathematical analysis like technical analysis and fundamental analysis. Such techniques do not use linguistic clues like the words "new release", "develop", "apology", and "abuse" which are likely to affect on stock prices.

While inexpert people generally have disadvantage in the amount of information they can obtain to make a decision than professional investors news articles are popular sources of information for many people because they are published everyday (i.e., they provide fresh information) and easily accessed. They include many (good or bad) news about companies that seem to have effects on their stock prices. Therefore, it is important to investigate the challenges and opportunities in extracting useful information to predict stock prices from news articles.

Watanabe et al.[7] proposed a system to predict whether the stock value will increase or decrease based on the analysis of news articles categorized by their themes. Their study was limited to some restricted set of texts while our analysis consider all given texts.

3.1 Method of Analysis

The corpus analyzed in this research was Mainichi News Articles 98–99[4]. We also use Japanese market price data by PanRolling[8]. We selected three genres of companies: electronics companies, automakers, and beverage manufacturers. Articles that include each company name in their headlines were selected for analysis. Morphological analysis was performed on each article and nouns, verbs, adjectives, and unknown words were extracted from it.[4]

News articles were categorized according to the changes in the corresponding stock prices. One category was *change to decrease* and the other was *change to increase*. The former is related to stock prices that were increasing or stable in the previous days and decreasing in the following days. The latter is related to ones that were decreasing or stable in the previous days and increasing in the following days. We set a threshold value T to decide whether the value was increasing, decreasing or stable. T is defined as

$$T = \frac{n}{a}$$

where n is the average stock price for each genre and a is a parameter to adjust.

We calculated the semantic orientation (GOOD/BAD) of each word by the following way.

1. Calculate the function of stock changes in the preceding (X_{pre}) and following (X_{next}) days by the following formula.

$$X_{pre} = SP_{-1} - SP_{-3}$$

$$X_{next} = SP_2 - SP_0.$$

Here, SP_n is the average stock price in the target genre on the n-th day ($n = 0$ is the target day). X_{pre} is categorized into *decreasing* if $X_{pre} \leq -T$, *increasing* if $X_{pre} \geq T$, and *stable* otherwise. X_{next} is categorized in the same way. After that, each day is categorized *change to decrease* if X_{pre} is *increasing* or *stable* and X_{next} is *decreasing*, or *change to increase* if X_{pre} is *decreasing* or *stable* and X_{next} is *increasing*.

2. Caluclate $\overline{TF}(i)$, which is the score for the word i, in the following formula.

$$\overline{TF}(i) = \frac{\sum_{d \in D} TF_{norm}(i,d)}{DF(i)}$$

where $TF_{norm}(i,d)$, the term frequency normalized by the document length, is defined as

$$TF_{norm}(i,d) = \frac{tf(i,d)}{\sum_{w \in d} tf(w,d)}.$$

Here, D is the set of all documents and $DF(i)$ is the number of documents that contains the word i. d and w stand for a document and a word, respectively.

[4] If two or more nouns were adjacent, they were extracted as one noun compound.

Table 5. The number of *appropriate* words in each category.

	1.0	0.9	0.8	0.7	0.6	0.5
Electronics	8	(7)	12	7	8	5
Automaker	7	(4)	9	10	5	5
Beverage	(5)	(0)	(3)	(4)	(1)	(5)
Total	15	(11)	21	17	13	10

3. Calculate the positive/negative ratio (C_{up}, C_{down}) by the following formula.

$$C_{up}(i) = \frac{\overline{TF}_{up}(i)}{\overline{TF}_{up}(i) + \overline{TF}_{down}(i)}$$

$$C_{down}(i) = \frac{\overline{TF}_{down}(i)}{\overline{TF}_{up}(i) + \overline{TF}_{down}(i)}$$

where $\overline{TF}_{up}(i)$ is the $\overline{TF}(i)$ value calculated on the articles on *change to increase* days and $\overline{TF}_{down}(i)$ is the $\overline{TF}(i)$ value calculated on the articles on *change to decrease* days.

We tested various a values ($a = 10, 15, 20, 25, 30$) and selected $a = 30$ as the best case. Each word was categorized by the range of its C_{up} or C_{down} value (the larger one of C_{up} and C_{down} was selected as the value for each word) into 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. (0.5 means that $0.5 \leq C < 0.6$, and so on. 1.0 means $C = 1.0$.) We obtained 100 words for each category and manually checked if C_{up} and C_{down} values are appropriate or not (*i.e.,* $C_{up} > C_{down}$ for positive words and $C_{down} > C_{up}$ for negative words).

Table 5 shows the number of *appropriate* words in each category.[5] We observed that the category 0.8 contains the largest number of appropriate words, and the number decreased as proceeding to 0.7, 0.6, and 0.5. This result indicates that the C_{up} and C_{down} values we defined have some correlation with the actual semantic orientations of the words. Some example words from the category 0.8 is shown in Table 6.

3.2 Conclusions and Future Work

In this section, we reported our attempt to analyze relations between words and changes of stock prices. We proposed a method to calculate the indicate value to reflect impression of each word. We observed that the calculated values have some correlation with actual semantic orientations. Future work includes the extraction of more advanced features like dependency relations or bigrams that will be more useful for stock price prediction.

[5]Numbers put in brackets () indicate that the number of words in the category was lower than 100. These numbers were not used in the calculation of total numbers.

4 System-3: A System for Graph Generation from News Articles

We developed a system to extract a set of graphs from a given news articles. The input to the system is a set of news articles[6] and the output is a set of graphs. The system participated in the MuST T2N task.

4.1 Graph Generation

Our system is based on the attribute-name extraction system described in Section 2. It extracts attribute names from given news articles with SVMs trained on the MuST corpus. (New corpus provided in the T2N task were not used both for SVM training and for construction of rules to extract information needed for graph generation, which is described later.) Graphs are generated by using the Java JFreeChart class[2] after attributes and their values are automatically extracted and clustered. Figure 2 shows the workflow of our system.

It is not enough to use the System-1 because it only extracts attribute names. There are some problems to solve for drawing a graph automatically. We discuss these problems in the rest of this subsection.

Value Extraction. First, information related to each attribute name including its value must be extracted. Given a set of attribute names and a set of unit names, the system performs the following processes for each (attribute, unit) pair. First, each sentence in the target corpus is parsed and the bunsetsus similar to the attribute name is extracted. Then, the system searches for the value of the extracted bunsetsu by traversing the dependency tree. The value must contain numbers and the given unit name. In addition, the bunsetsu directly depending on the attribute is extracted as the *modifier for the attribute* and the bunsetsu directly depending on the value is extracted as the *modifier for the value.*

Time Point Extraction. Second, time point for each attribute name and value must be extracted. The time point for each attribute (and its value) is estimated by using heuristic rules that use the time stamp of the document and time expressions like "* gatsu (year)", "last year", "last month", etc. found in the modifier for the attribute and the modifier for the value.

Attribute Clustering. Third, extracted attribute names are related to each other for inclusion in the same graph. It is not trivial because of existence of synonymous attribute names (*i.e.,*different

[6]We only tested the case where this set was a set of documents from one of MuST categories like "documents about Gasoline" or "documents about PC", although the system can accept other type of document sets.

Table 6. Examples of words from the category 0.8**.**

Genre	Orientation	Example	$\overline{TF}_{up}$	$\overline{TF}_{down}$
Electronics	GOOD(0.89)	製造 (manufacturing)	0.01185	0.00143
	GOOD(0.87)	安心 (reassuring)	0.01163	0.00160
Automaker	GOOD(0.85)	輸出 (export)	0.01600	0.00292
	GOOD(0.83)	新型 (new model)	0.00354	0.00071
Electronics	BAD(0.88)	回収 (recall)	0.00161	0.01178
	BAD(0.87)	大幅減益 (significant drop in income)	0.00240	0.01566
Automaker	BAD(0.88)	閉鎖 (disuse)	0.00069	0.00498

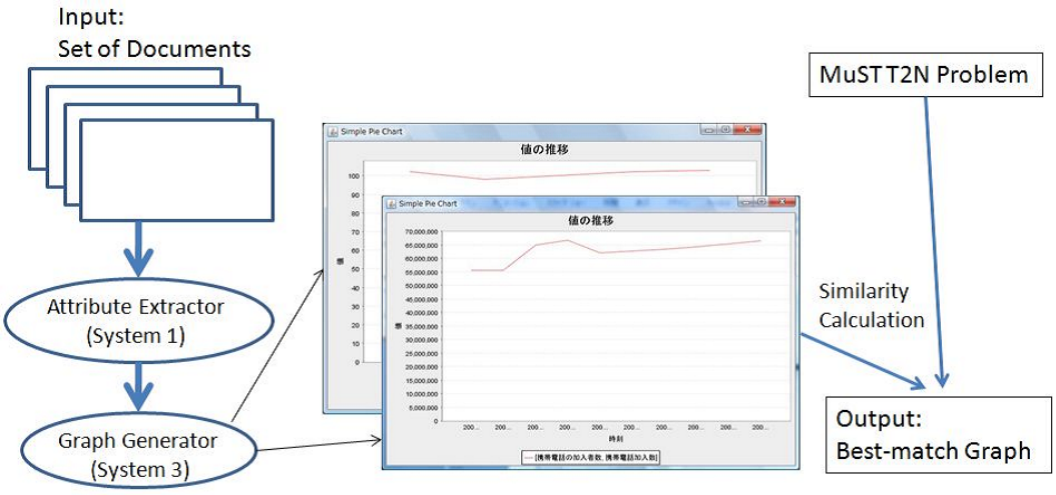

Figure 2. System Workflow for MuST T2N Task.

attribute names may have the same meaning). We solved this problem by calculating similarities between $(attribute, value)$ pairs. If the similarity between two attribute names (calculated as the number of common characters among two attribute names) were below the threshold[7], the similarity between $(attribute, value)$ pairs is defined as zero. Otherwise, the similarity is defined as the ratio (of the smaller to the larger) between the two values. This definition is based on the intuition that values in the same attribute do not show drastic changes. Currently, the threshold value is set to 0.5. Therefore, if one value is larger than the half of the other value (and the former is smaller than the latter), the two values are linked to be in the same attribute regardless of their time points. Two sets of values largely different from each other, such as relative values and absolute values of the same attribute, can be discriminated well by using this method. Our clustering algorithm activates edges (links) in (descending) order of the similarity values. If two components connected by the edge have one or more common dates, the edge is not activated based on the assumption that *two values from the same date must be different*. The result of clustering is a set of components connected by the activated links.

[7] currently set to 3

4.2 Participation in the MuST T2N task

As described in the previous subsection, the system generates a set of graphs from the given news articles. To solve problems from the T2N task, it is needed to select the most appropriate graph from the generated ones. The system calculated *similarities between each graph and each problem*, and output the most similar graph as a solution to each problem.

We define the similarity between a graph and a problem by the similarities between attribute names in the graph and the attribute name in the problem. It is defined as the *sum* of similarities between each attribute name in the graph and the attribute name in the problem. This definition (defined not by the *average* but by the *sum*) is based on the observation that taking the average results in the selection of the graph with the small number of (attribute, value) pairs where one or two attributes are occasionally very similar to the attribute name in the problem.

Table 7 shows the results of the T2N task. The system did not output any graph for problems 010501–010503 because values in these problems have no unit

Table 7. T2N Evaluation Results

Problem	Precision	Recall	F-measure
MuSTT2N010101	60.0	18.8	28.6
MuSTT2N010102	0.0	0.0	0.0
MuSTT2N010201	0.0	0.0	0.0
MuSTT2N010202	33.3	13.3	19.0
MuSTT2N010301	54.5	31.6	40.0
MuSTT2N010302	0.0	0.0	0.0
MuSTT2N010303	11.1	5.9	7.7
MuSTT2N010304	22.2	50.0	30.8
MuSTT2N010401	28.6	7.4	11.8
MuSTT2N010402	0.0	0.0	0.0
MuSTT2N010403	0.0	0.0	0.0
MuSTT2N010404	0.0	0.0	0.0
MuSTT2N010501	–	–	–
MuSTT2N010502	–	–	–
MuSTT2N010503	–	–	–
MuSTT2N010601	50.0	11.5	18.8
MuSTT2N010602	40.0	22.2	28.6
MuSTT2N010603	0.0	0.0	0.0
MuSTT2N010604	20.0	20.0	20.0
MuSTT2N010701	0.0	0.0	0.0
MuSTT2N010702	66.7	28.6	40.0
MuSTT2N010801	0.0	0.0	0.0
MuSTT2N010802	80.0	40.0	53.3
MuSTT2N010803	0.0	0.0	0.0
MuSTT2N010804	0.0	0.0	0.0

names, which is necessary for our current system. We observed that the result shows the tendency that there were mixed cases of successes and failures in the same document set. For example, extraction succeeded to some extent for the problem 010401, but extraction totally failed for the problems 010402–010404. In these problems, four similar attributes (cabinet's approval rating, cabinet's disapproval rating, LDP's approval rating, Democrats' approval rating) must be distinguished from each other. The ability of our system to discriminate such similar attributes (in their values) was not enough for these tasks.

We show an example result for Gasoline topic in table 8. Here, each cluster is sorted according to the similarity to the problem MuSTT2N010101. Clusters with similarities below 3.0 are omitted[8]. We observed that some Gasoline Price values were clustered and ranked to top-1 correctly. However, 5 Gasoline Price values (including "bottom of Gasoline Price") went to the second cluster where one noise ("Diesel Price") drew relatively low Gasoline Price values. This result suggests that our algorithm for attribute clustering that simply uses the number of common characters leaves room for improvement. We plan to exclude these noises by, for example, finding "unique characters" that can robustly make distinctions between different attributes. Gasoline Price values in the second cluster were mentioned as the past values (when the prices were relatively low) like *last year spring* and *one half years ago*. The system failed to extract most of these dates correctly, which suggests that time extraction rules in our algorithm should be also improved.

[8]Other 8 clusters (all of them were size-1 clusters) included Heating Oil Price Per 18 Litter, Increase of Electric Bill and Gas Fee, etc.

4.3 Conclusions and Future Work

We proposed a system to generate graphs automatically from a given text set. After extracting attribute-names with SVMs as described in Section 2, the system performs value extraction, time-point extraction, and attribute-value clustering needed for graph generation. We also developed an algorithm to automatically find a graph that match the best with the problem from the MuST T2N task. Evaluation results in the T2N task suggested that our system still leave room for improvement in discrimination between value-similar attributes.

Future work includes developing fully-automatic extraction algorithms because the current system requires unit names to be extracted as values. Addressing the problem of no-unit values are also important. We also plan to extract relative expressions like `5% increase` to make the graphs more reliable.

5 Summary

We reported our systems for MuST: the attribute name extraction system, the text-based stock price analysis system, and the automatic graph generation system. The attribute name extraction system was developed based on the SVMs trained on the MuST corpus. We developed the automatic graph generation system based on the attribute name extraction system to participate in the MuST T2N task. We observed that the system can solve problems from the T2N task to some extent by calculating similarities between generated graphs and T2N problems.

References

[1] M. Asahara and Y. Matsumoto. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 21–27, 2000.

[2] JFreeChart. http://www.jfree.org/jfreechart/.

[3] T. Kudo and Y. Matsumoto. Japanese dependency analyisis using cascaded chunking. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 63–69, 2002.

[4] Mainichi-Newspaper-Publishing-Company. *CD-Mainichi Newspapers*. NICHIGAI ASSOCIATES, INC., 1998–1999.

Table 8. Detailed Clustering Results for "Yen" Values from Gasoline-Related Articles. "dupli" means the same-date value is included in higher-rank clusters. GP stands for "(Regular) Gasoline Price."

No.	Description	Values	Size	Similarity
1	GP	98.0–103.0	6	36.0
2	Diesel Price (83.0), Bottom of GP	83.0–92.0	6	28.0
3	Tax in GP (60.0), GP (dupli.)	60.0–103.0	3	18.0
4	Increase of GP	2.0–2.5	3	17.0
5	Electric Bill (200.0), GP	105.0–200.0	3	10.0
6	Increase of Heating Oil Price or GP	1.0–1.2	2	10.0
7	Increase of Crude Oil Price, Un-shifted Cost in Gas Oil Price	9.0–15.0	3	9.0
8	Increase of GP	11.0	1	9.0
9	Increase of GP (in Gulf Crisis)	12.0	1	8.0
10	Increase of GP	1.0	1	8.0
11	Increase of Crude Oil Price	15.5–16.0	2	6.0
12	Un-shifted Cost in GP	4.0	1	4.0
13	GP (dupli.)	104.0	1	4.0
14	Heating Oil Price	46.4	1	4.0
15	GP (dupli.)	103.0	1	4.0
16	Increase of GP, Increase of Heating Oil Price (per 18 litter)	3.0–6.0	2	3.0

[5] H. Nakagawa. Automatic term recognition based on statistics of compound nouns. *Terminology*, 6(2):195–210, 2000.

[6] K. Nakano and Y. Hirai. Japanese named entity extraction with bunsetsu features. *Transactions of Information Processing Society of Japan*, 45(3):934–941, 2004. (in Japanese).

[7] T. Ogawa and I. Watanabe. Mining of stock prices and news articles. *IPSJ SIG Notes*, 2001(20):137–144, 2001. (in Japanese).

[8] PanRolling. http://www.panrolling.com/.

[9] T. Sugiura, M. Yoshida, K. Yamada, H. Masuda, and H. Nakagawa. Mining news articles by numbers. In *Proceedings of the Sixth Forum on Information Technology (FIT 2007)*, pages 161–164, 2007. (in Japanese).

[10] K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, and H. Isahara. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 326–335, 2000.

[11] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.