Looking for Trouble

Stijn De Saeger **Kentaro Torisawa**
Language Infrastructure Group
National Institute of Information
and Communications Technology
{stijn,torisawa}@nict.go.jp

Jun'ichi Kazama
School of Information Science
Japan Advanced Institute
of Science and Technology
kazama@jaist.ac.jp

Abstract

This paper presents a method for mining potential *troubles* or *obstacles* related to the use of a given *object*. Some example instances of this relation are ⟨*medicine*, *side effect*⟩ and ⟨*amusement park*, *height restriction*⟩. Our acquisition method consists of three steps. First, we use an unsupervised method to collect training samples from Web documents. Second, a set of expressions generally referring to troubles is acquired by a supervised learning method. Finally, the acquired troubles are associated with objects so that each of the resulting pairs consists of an object and a trouble or obstacle in *using* that object. To show the effectiveness of our method we conducted experiments using a large collection of Japanese Web documents for acquisition. Experimental results show an 85.5% precision for the top 10,000 acquired troubles, and a 74% precision for the top 10% of over 60,000 acquired object-trouble pairs.

1 Introduction

The Stanford Encyclopedia of Philosophy defines an artifact as "*…an object that has been intentionally made or produced for a certain purpose*". Because of this purpose-orientedness, most human actions relating to an object or artifact fall into two broad categories — actions relating to its intended use (e.g. *reading a book*), and the preparations necessary therefore (like *buying the book*). Information concerning potential obstacles, harmful effects or troubles that interfere with this intended use is therefore highly relevant to the user. While some such troubles are self-evident, others represent a genuine obstacle whose existence was thusfar unknown to the user. For example, in early 2008 a food poisoning case caused a big media stir in Japan when dozens of people fell ill after eating Chinese-imported frozen food products containing residual traces of toxic pesticides. While supposedly the presence of toxic chemicals in imported frozen foods had already been established on several occasions before, until the recent incidents public awareness of these facts remained low. In retrospect, a publicly available system suggesting "residual agrichemicals" as a potential danger with the consumption of "frozen foods" based on information mined from a large collection of Web documents might have led to earlier detection of this crisis. From the viewpoint of manufacturers as well, regularly monitoring the Internet for product names and associated troubles may allow them to find out about perceived flaws in their products sooner and avoid large scale recalls and damage to their brand.

For a less dramatic example, searching for *"Tokyo Disneyland"* on the Internet typically yields many commercial sites offering travel deals, but little or no information about potential obstacles such as "height restrictions" (constraints on who can enjoy a given attraction[1]) and "traffic jams" (a necessary preparation for enjoying a theme park is actually getting there in time). Ofter users have no way of finding out about this until they actually go there.

These examples demonstrate the importance of a highly accurate automatic method for acquiring what we will call "object-trouble" relations — pairs $\langle e_o, e_t \rangle$ in which the thing referred to by e_t constitutes an (actual or potential) trouble, obstacle or risk in the *context of use* of an object e_o.

[1] For example, one has to be over 3 ft. tall to get on the Splash Mountain.

Large scale acquisition of this type of contextual knowledge has not been thoroughly studied so far. In this paper, we propose a method for automatically acquiring Japanese noun phrases referring to troubles, (henceforth referred to as *trouble expressions*), and associating them with expressions denoting artifacts, objects or facilities.

Our acquisition method consists of three steps. As a first step, we use an unsupervised method for efficiently collecting training data from a Web corpus. Then, a set of expressions denoting troubles is acquired by a supervised learning method — Support Vector Machines (Vapnik, 1998) — trained on this data. Finally, the acquired trouble expressions are paired with noun phrases referring to objects, using a combination of pairwise mutual information and a verb-noun dependency filter based on statistics in a Web corpus.

A broad focus on noun-verb dependencies — and in particular the distinction between dependency relations with *negated* versus *non-negated* verbs – is the main characteristic of our method. While this distinction did not prove useful for improving the supervised classifier's performance in step 2, it forms the basis underlying the unsupervised method for training sample selection in the first step, and the final filtering mechanism in the third step.

The rest of this paper is organized as follows. Section 2 points out related work. Section 3 examines the notion of trouble expressions and their evidences. Section 4 describes our method, whose experimental results are discussed in Section 5.

2 Related Work

Our goal of automatically acquiring object-trouble pairs from Web documents is perhaps best viewed as a problem of semantic relation extraction. Recently the Automatic Content Extraction (ACE) program (Doddington et al., 2004) is a well-known benchmark task concerned with the automatic recognition of semantic relations from unstructured text. Typical target relations include "Reaction" and "Production" (Pantel and Pennacchiootti, 2006), "person-affiliation" and "organization-location" (Zelenko et al., 2002), "part-whole" (Berland and Charniak, 1999; Girju et al., 2006) and temporal precedence relations between events (Chklovski and Pantel, 2004; Torisawa, 2006). Our current task of acquiring "object-trouble" relations is new and object-trouble relations are inherently more abstract and indirect than relations like "person-affiliation" — they crucially depend on additional knowledge about *whether* and *how* a given object's use might be hampered by a specific trouble.

Another line of research closely related to our work is the recognition of semantic orientation and sentiment analysis (Turney, 2002; Takamura et al., 2006; Kaji and Kitsuregawa, 2006). Clearly troubles should be associated with a negative orientation of an expression, but studies on the acquisition of semantic orientation traditionally do not bother with the context of evaluation. While recent work on sentiment analysis has started to associate sentiment-related attribute-evaluation pairs to objects (Kobayashi et al., 2007), these attributes usually concern *intrinsic* properties of the objects, such as a digital camera's colors — they do not extend to sentiment-related factors *external* to the object like "traffic jams" for theme parks. The acquisition method proposed in this work addresses both these matters.

Finally, our task of acquiring trouble expressions can be regarded as hyponymy acquisition, where target expressions are hyponyms of the word "trouble". Although we used the classical lexico-syntactic patterns for hyponymy acquisition (Hearst, 1992; Imasumi, 2001; Ando et al., 2003) to reflect this intuition, our experiments show we were unable to attain satisfactory performance using lexico-syntactic patterns alone. Thus, we also use verb-noun dependencies as evidence in learning (Pantel and Ravichandran, 2004; Shinzato and Torisawa, 2004). We treat the evidences uniformly as elements in a feature vector given to a supervised learning method, which allowed us to extract a considerably larger number of trouble expressions than could be acquired by sparse lexico-syntactic patterns alone, while still keeping decent precision. What kind of hyponymy relations can be acquired by noun-verb dependencies is still an open question in NLP. In this work we show that at least trouble expressions can successfully be acquired based on noun-verb dependency information alone.

3 Trouble Expressions and Features for Their Acquisition

In section 1 we have characterized trouble expressions as a kind of "trouble" that occurs in the specific context of using some object, in other words:

1. *hyponym* ni nita *hypernym*
   (*hyponym* similar to *hypernym*)
2. *hyponym* to yobareru *hypernym*
   (*hypernym* called *hyponym*)
3. *hyponym* igai no *hypernym*
   (*hypernym* other than *hyponym*)
4. *hyponym* no youna *hypernym*
   (*hypernym* like *hyponym*)
5. *hyponym* to iu *hypernym*
   (*hypernym* called *hyponym*)
6. *hyponym* nado(no|,) *hypernym*
   (*hypernym* such as *hyponym*)

Table 1: Japanese lexico-syntactic patterns for hyponymy relations

as hyponyms of "trouble". Hence one source of evidence for acquisition are hyponymy relations with "trouble" or its synonyms. Another characterization of trouble expressions is to think of them as obstacles in a broad sense: things that prevent certain actions from being undertaken properly. In this sense *traffic jams* and *sickness* are troubles since they prevent people from going places and doing things. This assumption underlies a second important class of evidences for learning.

More precisely, the evidence used for learning is classified into three categories: (i) lexico-syntactic patterns for hyponymy relations, (ii) dependency relations between expressions and negated verbs, and (iii) dependency relations between expressions and non-negated verbs. The first two categories are assumed to contain positive evidence of trouble expressions, while we assumed the third to function mostly as negative evidence. Our experiments show that (i) turns out to be less useful than expected, while the combination of (ii) and (iii) alone already gave quite reasonable precision in acquiring trouble expressions. Each category of evidence is described further below.

3.1 Lexico-syntactic patterns for hyponymy

Since trouble expressions are hyponyms of "trouble", one obvious way of acquiring trouble expressions is to use classical lexico-syntactic patterns for hyponymy acquisition (Hearst, 1992). Table 1 lists some of the patterns proposed in studies on hyponymy acquisition for Japanese (Ando et al., 2003; Imasumi, 2001) that are utilized in this work.

In actual acquisition, we instantiated the *hypernym* positions in the patterns by Japanese translations of "trouble" and its synonyms, namely toraburu (troubles), sainan (accidents), saigai (disasters) and shougai (obstacles or handicaps), and used the instantiated patterns as evidence. Hereafter, we call these patterns **LSPHs** (Lexico-Syntactic Patterns for Hyponymy).

3.2 Dependency relations with Verbs

We expect expressions that frequently refer to troubles to have a distinct *dependency profile*, by which we mean a specific set of dependency relations with verbs (i.e. occurrences in specific *argument positions*). If T is a trouble expression, then given a sufficiently large corpus one would expect to find a reasonable number of instantiations of patterns like the following:

- T kept X from doing Y.
- X didn't enjoy Y because of T.

Similarly, "X enjoyed T" would present negative evidence for T being a trouble expression. Rather than single out a set of particular dependency relations suspected to be indicative of trouble expressions, we let a supervised classifier learn an appropriate weight for each feature in a large vector of dependency relations. Two classes of dependency relations proved to be especially beneficial in determining trouble candidates in an unsupervised manner, so we discuss them in more detail below.

Dependency relations with negated verbs Following our characterization of troubles as things that *prevent* specific actions from taking place, we expect a good deal of trouble expressions to appear in patterns like the following:

- X cannot go to Y *because of* T.
- X did not enjoy Y *because of* T.

The important points in the above are (i) the negated verbs and (ii) the mention of T as the reason for not verb-ing. The following are Japanese translations of the above patterns. Here P denotes postpositions (Japanese case markers), V stands for verbs and the phrase "because of" is translated as the postposition de.

- T de Y ni ikenai.
  P P V (cannot go)
- T de X ga tanoshikunakatta.
  P P V (did not enjoy)

We refer to the following dependency relations between expressions marked with the postposition `de` and negated verbs in these patterns as **DNVs** (Dependencies to Negated Verbs).

$$T \ \texttt{de} \rightarrow \text{negated verb} \quad (1)$$

We allow any verb to be the negated verb, expecting that inappropriate verbs will be less weighted by machine learning techniques. For instance, the dependency relations to negated verbs with an originally negative orientation such as "suffer" and "die" will not work as positive examples for trouble expressions.

Unfortunately, these patterns still present only *weak* evidence for trouble expressions. The precision of the trouble expressions collected using DNV patterns is extremely low — around 6.5%. This is due to the postposition `de`'s ambiguity — besides "because of" relations it also functions as a marker for location, time and instrument relations, among others. As a result, non-trouble expressions such as "by car" (instrument) and "in Tokyo" (location) are marked by the postposition `de` as well.

We consider a second class of dependency relations, acting mostly as a counter to the noisy expressions introduced by the ambiguity of the postposition `de`.

Dependency relations with non-negated verbs The final type of evidence is formulated as the following dependency relation.

$$T \ \texttt{de} \rightarrow \text{non-negated verb}$$

We call this type of relation **DAVs** (Dependencies to Affirmative Verbs). The use of these patterns is motivated by the intuition that noisy expressions found with DNVs, such as expressions about locations or instruments, will also frequently appear with non-negated verbs. That is, if you observe "cannot go to Y (by / because of) X" and X is not a trouble expression, then you can expect to find "can go to Y (by / because of) X" as well.

Our initial expectation was that the DNV and DAV evidences observed with the postposition `de` alone would contain sufficient information to obtain an accurate classifier, but this expectation was not borne out by our early experiments. As it turns out, using dependency relations to verbs in *all* argument positions as features to the SVM resulted roughly in a 10$\sim$15% increase in precision. Therefore in our final experiments we let the DNV and DAV evidence consist of dependencies with four additional postpositions (`ha`, `ga`, `wo` and `ni`), which are used to indicate topicalization, subject, object and indirect object. We found that the SVM was quite successful in learning a dependency profile for trouble expressions based on this information.

Nonetheless, the DNV/DAV patterns proved to be useful besides as evidence for supervised learning, for instance in gathering sufficient trouble candidates and sample selection when preparing training data [2].

4 Method

As mentioned, our method for finding troubles in using some objects consists of three steps, described in more detail below.

Step 1 Gather training data with a sufficient amount of positive samples using an unsupervised method to reduce the workload of manual annotation.

Step 2 Collect expressions commonly perceived as troubles by using the evidences described in the previous section.

Step 3 Identify pairs of trouble expressions and objects such that the trouble expressions represent an obstacle in using the objects.

4.1 Step 1: Gathering Training Data

We considered noun phrases observed with the LSPH and DNV evidences as candidate trouble expressions. However, we still found only 7% of the samples observed with these evidences to be real troubles. Because of the diversity of our evidences (dependencies with verbs) we need a reasonable amount of positive samples in order to obtain an accurate classifier. Without some sample selection scheme, we would have to manually annotate about 8000 samples in order to obtain only 560 positive samples in the training data. For this reason we used the following scoring function as an unsupervised method for sample selection.

$$Score(e) = \frac{f_{\text{LSPH}}(e) + f_{\text{DNV}}(e)}{f_{\text{LSPH}}(e) + f_{\text{DNV}}(e) + f_{\text{DAV}}(e)} \quad (2)$$

Here, $f_{\text{LSPH}}(e)$, $f_{\text{DNV}}(e)$ and $f_{\text{DAV}}(e)$ are the frequencies that expression e appears with the respective evidences. Intuitively, this function gives

[2] We will discuss yet another use of the DNV evidence in step 2 of our acquisition method.

a large score to expressions that occur frequently with the positive evidences for trouble expressions (LSPHs and DNVs), or those that appear rarely with the negative evidences (DAVs). In preparing training data we ranked all candidates according to the above score, and annotated N elements from the top and bottom of the ranking as training data. In our experiments, the top elements included a reasonable number of positive samples (25.8%) while there were almost none in the worst elements.

4.2 Step 2: Finding Trouble Expressions

In this step our aim is to acquire expressions often associated with troubles. We use a supervised classifier, namely Support Vector Machines (SVMs) (Vapnik, 1998) for distinguishing troubles from non-troubles, based on the evidences described above. Each dimension of the feature vector presented to the SVM corresponds to the observation of a particular evidence (i.e., these are binary features). We tried using frequencies instead of binary feature values but could not find any significant improvement in performance. After learning we sort the candidate trouble expressions according to their distance to the hyperplane learned by the SVM, and consider the top N expressions in the sorted list as true trouble expressions.

4.3 Step 2: Identifying Object-Trouble Pairs

In this third stage we rank possible combinations of objects and trouble expressions acquired in the previous step according to their degree of association and apply a filter using negated verbs to the top pairs in the ranking. The final output of our method is the top N pairs that survived the filtering. We describe each step below.

Generating Object-Trouble Pairs To generate and rank object-trouble pairs we use a variant of pairwise mutual information that scores an object-trouble pair $\langle e_o, e_t \rangle$ based on the observed frequency of the following pattern.

$$\begin{array}{ccc} e_o & \texttt{no} & e_t \\ & \text{P} & \end{array} \qquad (3)$$

The postposition no is a genitive case marker, and the whole pattern can be translated as "e_t of / in e_o". We assume that appearance of expression e_t in this pattern refers to a trouble in using the object e_o.

More precisely, we generate all possible combinations of trouble expression and objects and rank them according to the following score.

$$I(e_o, e_t) = \frac{f(\text{“}e_o \; \texttt{no} \; e_t\text{”})}{f(\text{“}e_o\text{”})f(\text{“}e_t\text{”})} \qquad (4)$$

where $f(e)$ denotes an expression e's frequency. This score is large when the pattern "e_o no e_t" is observed more frequently than can be expected from e_o and e_t's individual frequencies. Frequency data for all noun phrases was precomputed for the whole Web corpus.

Filtering Object-Trouble Pairs The filtering in the second step is based on the following assumption.

Assumption If a trouble expression e_t refers to a trouble in using an object e_o , there is a verb v such that v frequently co-occurs with e_o and v has the following dependency relation with e_t.

e_t de → negated v

The intuition behind this assumption can be explained as follows. First, if e_o denotes an object or artifact then its frequently co-occurring verbs are likely to be related to a use of e_o. Second, if e_t is a trouble in using e_o, there is some action associated with e_o that e_t prevents or hinders, implying that e_t should be observed with its negation. For instance, if "traffic jam" is a trouble in using an amusement park, then we can expect the following pattern to appear also in a corpus.

juutai	de	yuuenchi	ni	ikenai.
traffic jam	P	*theme park*	P	V (cannot go)

cannot go to a theme park because of a traffic jam

The verb "to go" co-occurs often with the noun "theme park" and the above pattern contains the dependency relation "traffic jam de → cannot go". Substituting v in the hypothesis for "to go", the assumption becomes valid. Because of data sparseness the above pattern may not actually appear in the corpus, but even so the dependency relation "traffic jam de → cannot go" may be observed with other facilities, and thus making the assumption hold anyway.

As a final filtering procedure, we gathered K verbs most frequently co-occurring with each object and checked if the trouble expression in the pair has dependency relations with the K verbs in negated form and the postposition de. If none of the K verbs has such a dependency with the trouble expression, the pair is discarded. Otherwise, it is produced as the final output of our method.

5 Experimental Results

5.1 Finding Trouble Expressions

We extracted noun phrases observed in LSPH, DNV and DAV patterns from 6×10^9 sentences in about 10^8 crawled Japanese Web documents, and used the LSPH and DNV data [3] as candidate trouble expressions. After restricting the noun phrases to those observed more than 10 times in the evidences, we had 136,212 noun phrases. We denote this set as D. Extracting 200 random samples from D we found the ratio of troubles to non-troubles was around 7% and thus expected to find about $10,000$ real trouble expressions in D. [4]

Using the sample selection method described in Section 4.2 we prepared 6,500 annotated samples taken from D as training data. The top 3,500 samples included 912 positive samples and the worst 3,000 had just 9 positives, thereby confirming the effectiveness of the scoring function for selecting a reasonable amount of positive samples. Our final training data thus contained 14% positives.

For the feature vectors we included dependencies with all verbs occurring more than 30 times in our Web corpus. Besides the LSPH, DNV and DAV evidences discussed previously, we also included 10 additional binary features indicating for each of the five postpositions whether the expression was observed with DNV or DAV evidence *at all*, and found that including this information improved performance.

We trained a classifier with a polynomial kernel of degree 1 on these evidences using the software TinySVM[5], and evaluated the results obtained by the supervised acquisition method by asking three human raters whether a randomly selected sample expression denotes a kind of “trouble” in general situations. More specifically, we asked whether the expression is a kind of toraburu (trouble), sainan (accident), saigai (disaster) or shougai (obstacle or handicap).[6] For various combinations of evidences (described below), we presented 200 randomly sampled expressions from the top 10,000 expressions ranked according to the distance to the hyperplane learned by the SVM. Samples of all the compared methods are merged and shuffled before evaluation. The kappa statistic for assessing the inter-rater agreement was 0.78, indicating substantial agreement according to Landis and Koch, 1977.[7] We made no effort to remove samples used in training from the experiment, and found that the samples scored by the raters (1281 in total, after removal of duplicates) contained 67 training samples. The 200 samples from the “full” classifier contained 12 of these.

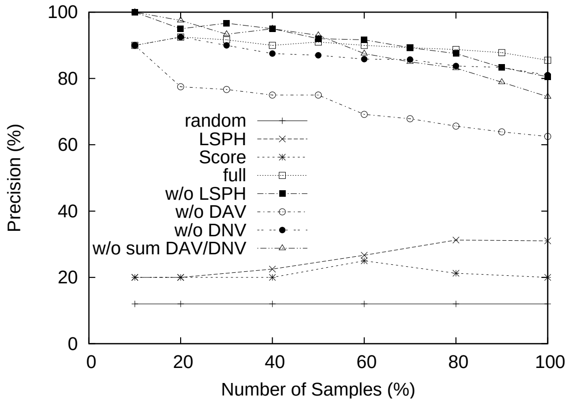

Figure 1: Performance of trouble expression acquisition (all 3 raters)

Fig. 1 shows the precision of the acquired trouble expressions compared to the samples labeled as troubles by all three raters. We sorted the samples according to their distance to the SVM hyperplane and plotted the precision of the top N samples. The best overall precision (85.5%) was obtained by a classifier trained on the full combination of evidences (labeled “full” in Fig. 1), maintaining over 90% precision for the top 70% of the 200 samples.

The remaining results show the relative contributions of the evidences. They were obtained by retraining the “full” classifier with a particular set of evidences removed, respectively LSPH evidences (labeled “w/o LSPH”), DNV evidences (“w/o DNV”), DAV evidences (“w/o DAV”) and the 10 features indicating the observation of

[3] We restricted noun phrases from the DNV data to those found with the postposition de, as these are most likely to refer to troubles.

[4] Thus, in the experiments we evaluated the top 10,000 samples output by our method.

[5] http://chasen.org/~taku/software/TinySVM/

[6] Actually one of the raters is a co-author of this paper, but he had no knowledge of the experimental setting nor had seen the acquired data prior to the experiment.

[7] This kappa value was calculated over the sum total of samples presented to the raters for scoring (duplicates removed).

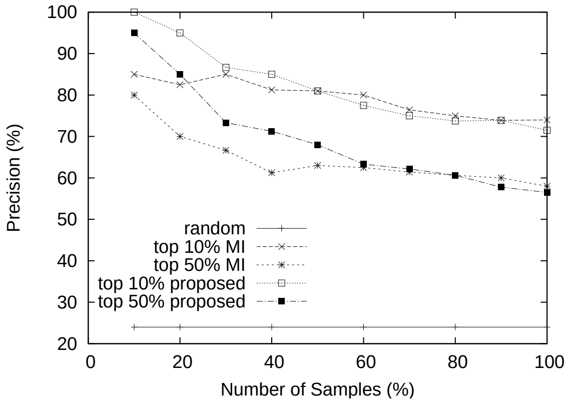

Figure 2: Performance of object-trouble pair acquisition (3 raters)

DAV/DNV evidence per postposition ("w/o sum DAV/DNV").

As Fig. 1 shows, leaving out DNV and even LSPH evidences did not affect performance as much as we expected, while leaving out the DAV dependencies gave more than 20% worse results. Of further interest is the importance of the binary features for DAV/DNV presence per postposition ("w/o sum DAV/DNV"). The absence of these 10 binary features accounts for a 10% precision loss compared to the full feature set (75%).

We also compared it with a baseline method using only lexico-syntactic patterns. We extracted 100 random noun phrases from the LSPH evidence in D for evaluation ("LSPH" in Fig. 1). The precision for this method was 31%, confirming that lexico-syntactic patterns for hyponymy constitute fairly weak evidence for predicting trouble expressions when used alone. "Score" shows the precision of the top 100 samples output by our $Score$ function from section 4. Finally, "random" (drawn as a straight line) denotes 100 random samples from D and roughly corresponds to our estimate of 7% true positives.

5.2 Identifying Object-Trouble Pairs

For the second step, we assumed the top 10,000 expressions obtained by our best-scoring supervised learning method ("full" in the previous experiments) to be trouble expressions, and proceeded to combine them with terms denoting artifacts or facilities.

We randomly picked 2,500 words that appeared as direct objects of the verbs `kau` ("to buy"), `tsukau` ("to use"), `tsukuru` ("to make"), `taberu` ("to eat") and `tanoshimu` ("to enjoy") more than 500 times in our Web corpus, assuming that this would yield a representative set of noun phrases denoting objects or artifacts.[8] Combining this set of objects with the acquired trouble expressions gave a list of 61,873 object-trouble pairs (all pairs $\langle e_o, e_t \rangle$ with at least one occurrence of the pattern "e_o `no` e_t"). Of this list, 58,570 pairs survived the DNV filtering step and form the final output of our method. For the DNV filtering, we used the top 30 verbs most frequently co-occurring with the object.

rank /raters	object	trouble expressions
1/3	`kousoku douro` (highway)	`sakeyoi unten` (drunk driving)
7/3	`kouseibushitsu` (antibiotics)	`ranyou` (abuse)
8/3	`suidousui` (tap water)	`suiatsu teika` (drop in water pressure)
21/3	`nouyaku` (agrichemicals)	`zanryuubushitsu` (residue)
98/2	`kikai` (machine)	`gohandan` (judgement error)
136/3	`zaisan` (estate)	`souzoku funsou` (succession dispute)

Figure 3: Examples of acquired object-trouble pairs

We again evaluated the resulting object-trouble pairs by asking three human raters whether the presented pairs consist of an object and an expression referring to an actual or potential trouble *in using* the object. The kappa statistic was 0.60, indicating *moderate* inter-rater agreement.

Fig. 2 shows the precision of the acquired pairs when comparing with what are considered true object-trouble relations by all three raters. Some examples of the pairs obtained by our method are listed in table 3 along with their ranking and the number of raters who judged the pair to be correct.

The precision for our proposed method when considering the top 10% of pairs ranked by the I score and filtered by the method described in section 4.3 is 71.5% ("top 10% proposed" in Fig. 2), which is actually worse than the results obtained *without* the final DNV filtering ("top 10% MI", 74%). For the first half of all samples however, we do observe some performance increase by the filtering, though both methods appear to converge in the second half of the graph. This tendency is mirrored closely when considering the results for the top 50% of all pairs (respectively "top 50% proposed" and "top 50% MI" in Fig. 2). The 15% decrease in precision compared to top 10% results

[8] We manually removed pronouns from this set.

indicates that performance drops gradually when moving to the lower ranked pairs.

6 Concluding Remarks and Future Work

We have presented an automatic method for finding potential troubles in using objects, mainly artifacts and facilities. Our method acquired 10,000 trouble expressions with 85.5% precision, and over 6000 pairs of objects and trouble expressions with 74% precision.

Currently, we are developing an Internet search engine frontend that issues warnings about potential troubles related to search keywords. Although we were able to acquire object-trouble pairs with reasonable precision, we plan to make a large-scale highly precise list of troubles by manually checking the output of our method. We expect such a list to lead to even more acurate object-trouble pair acquisition.

References

Ando, M., S. Sekine, and S. Ishizaki. 2003. Automatic extraction of hyponyms from newspaper using lexicosyntactic patterns. In *IPSJ SIG Technical Report 2003-NL-157*, pages 77–82. in Japanese.

Berland, M. and E. Charniak. 1999. Finding parts in very large corpora. In *Proc. of ACL-1999*, pages 57–64.

Chklovski, T. and P. Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proc. of EMNLP-04*.

Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *Proceedings of LREC 2004*, pages 837–840.

Girju, R., A. Badulescu, and D. Moldvan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING'92*, pages 539–545.

Imasumi, K. 2001. Automatic acqusition of hyponymy relations from coordinated noun phrases and appositions. Master's thesis, Kyushu Institute of Technology.

Kaji, N. and M. Kitsuregawa. 2006. Automatic construction of polarity-tagged corpus from html documents. In *Proc. of COLING/ACL 2006*, pages 452–459. (poster session).

Kobayashi, N., K. Inui, and Y. Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proc. of EMNLP-CoNLL 2007*, pages 1065–1074.

Pantel, P. and M. Pennacchiootti. 2006. Espresso: Leveranging generic patterns for automatically harvesting semantic relations. In *Proc. of COLING/ACL-06*, pages 113–120.

Pantel, P. and D. Ravichandran. 2004. Automatically labelling semantic classes. In *Proc. of HLT/NAACL-04*, pages 321–328.

Shinzato, K. and K. Torisawa. 2004. Acquiring hyponymy relations from web documents. In *HLT/NAACL-04*, pages 73–80.

Takamura, H., T. Inui, and M. Okumura. 2006. Latent variable models for semantic orientation of phrases. In *Proc. of EACL 2006*, pages 201–208.

Torisawa, K. 2006. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In Moore, R.C., J.A. Bilmes, J. Chu-Carroll, and M. Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics.

Turney, P. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proc. of ACL'02*, pages 417–424.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Zelenko, D., C. Aone, and A. Richardella. 2002. Kernel methods for relation extraction. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 71–78, Morristown, NJ, USA. Association for Computational Linguistics.