Term Aggregation: Mining Synonymous Expressions using Personal Stylistic Variations

Akiko Murakami Tetsuya Nasukawa
IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato, Kanagawa 242-8502, Japan
{akikom, nasukawa}@jp.ibm.com

Abstract

We present a text mining method for finding synonymous expressions based on the distributional hypothesis in a set of coherent corpora. This paper proposes a new methodology to improve the accuracy of a term aggregation system using each author's text as a coherent corpus. Our approach is based on the idea that one person tends to use one expression for one meaning. According to our assumption, most of the words with similar context features in each author's corpus tend not to be synonymous expressions. Our proposed method improves the accuracy of our term aggregation system, showing that our approach is successful.

1 Introduction

The replacement of words with a representative synonymous expression dramatically enhances text analysis systems. We developed a text mining system called TAKMI (Nasukawa, 2001) which can find valuable patterns and rules in text that indicate trends and significant features about specific topics using not only word frequency but also using predicate-argument pairs that indicate dependencies among terms. The dependency information helps to distinguish between sentences by their meaning. Here are some examples of sentences from a PC call center's logs, along with the extracted dependency pairs:

- *customer broke a tp*
  → `customer...break`,
  `break...tp`
- *end user broke a ThinkPad*
  → `end user...break`,
  `break...ThinkPad`

In these examples, "*customer*" and "*end user*" and "*tp*" and "*ThinkPad*" can be assumed to have the same meaning in terms of this analysis for the call center's operations. Thus, these two sentences have the same meaning, but the differences in expressions prevent us from recognizing their identity. The variety of synonymous expressions causes a lack of consistency in expressions. Other examples of synonymous expressions are:

customer = cu = cus = cust = end user = user = eu

Windows95 = Win95 = w95

One way to address this problem is by assigning canonical forms to synonymous expressions and variations of inconsistent expressions. The goal of this paper is to find those of synonymous expressions and variations of inconsistent expressions that can be replaced with a canonical form for text analysis. We call this operation "term aggregation". Term aggregation is different from general synonym finding. For instance, "customer" and "end user" may not be synonyms in general, but we recognize these words as "customer" in the context of a manufacturers' call center logs. Thus, the words we want to aggregate may not be synonyms, but their role in the sentences are the same in the target domain from the mining perspective. Yet, we can perform term aggregation using the same methods as in synonym finding, such as using word feature similarities.

There are several approaches for the automatic extraction of synonymous expressions, such as using word context features, but the results of such approaches tend to contain some antonymous expressions as noise. For instance, a system may extract "agent" as a synonymous expression for "customer", since they share the same feature of being human, and since both words appear as subjects of the same predicates, such as "talk", "watch", and "ask".

In general, it is difficult to distinguish synonymous expressions from antonymous expressions based on their context. However, if we have a coherent corpus, one in which the use of expressions is consistent for the same meaning, the words extracted from that corpus are guaranteed to have different meanings from each other.

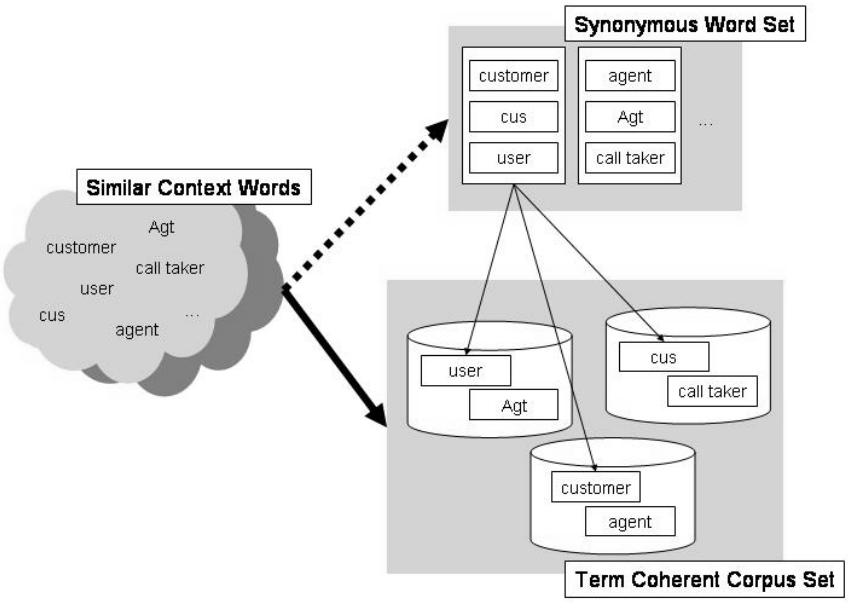

Figure 1: Synonym Extraction System using Coherent Corpus

Figure 1 illustrates the idea of such coherent corpora. Words with similar contexts within incoherent corpora consist of various expressions including synonyms and antonyms, as in the left hand side of this figure, because of the use of synonymous expressions as in the upper right box of the figure. In contrast, words with similar contexts within each coherent corpus do not contain synonymous expressions, as in the lower right box of the figure.

By using the information about non-synonymous expressions with similar contexts, we can deduce the synonymous expressions from the words with similar contexts within incoherent corpora by removing the non-synonymous expressions.

In this paper, we use a set of textual data written by the same author as a coherent corpus. Our assumption is that one person tends to use one expression to represent one meaning. For example, "user" for "customer" and "agt" for "agent" as in Figure 1.

Our method has three steps: extraction of synonymous expression candidates, extraction of noise candidates, and re-evaluation with these candidates. In order to evaluate the performance of our method, we conducted some experiments on extracting term aggregation sets. The experimental results indicate that our method leads to better precision than the basic synonym extraction approach, though the recall rates are slightly reduced.

The rest of this paper is organized as follows. First we describe the personal stylistic variations in each author's text in Section 2, and in Section 3 we will give an overview of our system. We will present the experimental results and discussion in Section 4. We review related work in Section 5 and we consider future work and conclude the paper in Section 6.

2 Personal Stylistic Variations in Each Authors' Corpora

According to our assumption, each author uses a unique expression to represent one semantic concept, even though various expressions can be used for representing the same meaning. To evaluate this assumption, we analyzed a call center's corpus, which was typed in by the call takers in a personal computer service call center [1].

Call Taker	A	B	C	D	E
customer	31	62	32	31	286
cust	6	335	2	3	2
eu	345	89	179	402	62
user	5	20	2	3	13

Table 1: The Variation of the Expressions for "customer" in each Call Taker's Text.

Table1 shows variations of the expressions for " customer" which were used by the call takers. This table shows that each call taker mainly used one

[1]The IBM PC Help Center

unique expression to represent one meaning with a consistency ratio of about 80%, but the other 20% are other expressions.

These results show our assumption holds for the tendency for one expression to have one meaning within the same author's corpus. However, it also demonstrated that multiple expressions for the same meaning appear within the same author's corpus even though the distribution of the appearences clearly leans toward one expression. Thus, we should consider this fact when we apply this assumption.

3 Experiments

3.1 Data Overview

In our experiments we used one month's worth of data stored in the call center, containing about five million words. The number of unique nouns was 29,961, and the number of unique verbs was 11,737, and 3,350,200 dependency pairs were extracted from the data. We then created ten subcorpora in such a manner that each of them contains data provided by the same call taker. The average number of predicate-argument pairs in each subcorpus was 37,454. In our experiments, we selected ten authors' corpus according to their size from the larger one.

To evaluate the experiments, we manually created some evaluation data sets. The evaluation data sets were made for ten target words, and the average number of variants was 7.8 words for each target word. Some examples are shown in Table2.

target concept	variants
customer	customer, cu, cus, cust, end user, user, eu
HDD	harddisk, hdd drive, HD, HDD, hdds, harddrive, hd, H.D
battery	Battery, batteyr, battery, battary, batt, bat
screen	display, monitor, moniter, Monitor

Table 2: Examples of Evaluation Data

For the cannonical expressions for each target word, we simply selected the most frequent expression from the variants.

3.2 Text Analysis Tool for Noisy Data

In the call center data there are some difficulties for natural language processing because the data contains a lot of informal writing. The major problems are;

- Words are often abbreviated
- There are many spelling errors
- Case is used inconsistently

Shallow processing is suitable for such noisy data, so we used a Markov-model-based tagger, essentially the same as the one described in (Charniak, 1993) in our experiments [2]. This tagger assigns a POS based on the distribution of the candidate POSs for each word and the probability of POS transitions extracted from a training corpus, and we used a manually annotated corpus of articles from the Wall Street Journal in the Penn Treebank corpus [3] as a training corpus. This tagger treats an unknown word that did not appear in the training corpus as a noun. In addition, it assigns a canonical form to words without inflections.

After POS tagging for each sequence of words in a document, it is possible to apply a cascaded set of rules, successively identifying more and more complex phrasal groups. Therefore, simple patterns will be identified as simple noun groups and verb groups, and these can be composed into a variety of complex NP configurations. At a still higher level, clause boundaries can be marked, and even (nominal) arguments for (verb) predicates can be identified. The accuracy of these analyses is lower than the accuracy of the POS assignment.

3.3 Term Aggregation using Personal Stylistic Variations

In this section we explain how to aggregate words using these word features. We have three steps for the term aggregation: creating noun feature vectors, extracting synonymous expressions and noise candidates, and a re-evaluation.

3.3.1 Creating Noun Feature Vectors

There is a number of research reports on word similarities, and the major approach is comparing their contexts in the texts. Contexts can be defined in two different ways: syntactic-based and window-based techniques. Syntactic-based techniques consider the linguistic information about part-of-speech categories and syntactic groupings/ relationships. Window-based techniques consider an arbitrary number of words around the given

[2]This shallow syntactic parser is called CCAT based on the TEXTRACT architecture (Neff, 2003) developed at IBM Watson Research Center.

[3]http:// www.cis.upenn.edu/ treebank/

rank	candidate
1	batt
2	batterie
3	bat
4	cover
5	BTY
6	batterry
7	adapter
8	bezel
9	cheque
10	screw

Table 3: battery's Synonymous Expression Candidates from the Entire Corpus

Author A		Author B	
rank	candidate	rank	candidate
1	battery	1	batt
2	controller	2	form
3	Cover	3	protector
4	APM	4	DISKETTE
5	screw	5	Mwave
6	mark	6	adapter
7	cheque	7	mouse
8	diskette	8	cheque
9	checkmark	9	checkmark
10	boot	10	process

Table 4: Noise Candidates from Each Author's Corpus

word. The words we want to aggregate for text analysis are not rigorous synonyms, but the "role" is the same, so we have to consider the syntactic relation based on the assumptions that words with the same role tend to modify or be modified by similar words (Hindle, 1990; Strzalkowski, 1992). On the other hand, window-based techniques are not suitable for our data, because the documents are written by several authors who have a variety of different writing styles (e.g. selecting different prepositions and articles). Therefore we consider only syntactic features: dependency pairs, which consist of nouns, verbs, and their relationships. A dependency pair is written as (noun, verb(with its relationship)) as in the following examples.

```
(customer, boot↓)
(customer, shut off↓)
(tp, shut off↑)
```

The symbol ↓ means the noun modifies the verb, and ↑ means the verb modifies the noun. By using these extracted pairs, we can assign a frequency value to each noun and verb as in a vector space model. We use a noun feature vector (NFV) to evaluate the similarities between nouns. The NFVs are made for each authors' corpora and for the entire corpus, which contains all of the author's corpora.

3.3.2 Extract Synonymous Expression Candidates and Noise Candidates

The similarity between two nouns that we used in our approach is defined as the cosine coefficient of the two NFVs. Then we can get the relevant candidate lists that are sorted by word similarities between nouns and the target word. The noun list from the entire corpus is based on the similarities between the target's NFV in the entire corpus and the NFVs in the entire corpus. These words are the synonymous expression candidates, which is the baseline system. The noun lists from the authors' corpora are extracted based on the similarities between the target's NFV in the entire corpus and the NFVs in each authors' corpora. The most similar word in an author's corpus is accepted as a synonymous expression for the target word, and the other similar words in the author's corpus are taken to not have the same meaning as the target word, even though the features are similar. These words are then taken

as the noise candidates, except for the most relevant words in each candidate list. If there are *N* authors, then *N* lists are extracted.

3.3.3 Re-evaluation

On the basis of our assumption, we propose a simple approach for re-evaluation: deleting the noise candidates in the synonymous expression candidates. However, as shown in Section 2, each author does not necessarily use only one expression for one meaning. For instance, while the call taker B in Table 1 mostly uses "cust", he/she also uses other expressions to a considerable degree. Accordingly if we try to delete all noise candidates, such synonymous expressions will be eliminated from the final result. To avoid this kind of over-deleting, we classified words into three types, "Absolute Term", "Candidate Term", and "Noise Candidate". First, we assigned the "Candidate Term" type to all of the extracted terms from the entire corpus. Second, the most relevant word extracted from each author's corpus was turned into an "Absolute Term". Third, the words extracted from all of the authors' corpora, except for the most relevant word in each author's corpus, were turned into the "Noise Candidate" type. In this step an "Absolute Term" does not change if the word is a noise candidate. Then the words listed as "Absolute Term" or "Candidate Term" are taken as the final results of the re-evaluation.

3.4 An Actual Example

In this section we will show an actual example of how our system works. In this example, the target word is "battery". First, the synonymous expression candidates are extracted from the entire corpus using the NFV of the target word in the entire corpus and the NFVs in the entire corpus. The relevant list is shown in Table 3. In this candidate list, we can find many synonymous expressions for "battery", such as "batt", "batterie", etc, however we also see some noise, such as "cover", "adapter", etc. In this step these words are tentavely assigned as "Candidate Term".

Second, the noise candidates are extracted from each authors' corpora by estimating the similarities between the target word's NFV in the entire corpus and the NFVs in the author's corpora. The noise candidate lists from two authors are shown in Table 4. The most relevant words in each author's corpora are "battery" and "batt", so the same words in the extracted "Candidate Term" list are turned into "Absolute Term" and remain undeleted even when "battery" and "batt" appear in the same author's corpus. The rest of the words in the noise candidate lists are noise, so the same words in the "Candidate Term" list are turned into "Noise Candidate", such as "cover", "adapter", "cheque", and "screw". Finally, we can get the term aggregation result as a list consisting of the words marked "Absolute Term" and "Candidate Term". The results are shown in Table 5.

batt
batterie
bat
BTY
batterry
bazel

Table 5: Results after Removing the Noise

4 Experimental Results and Discussion

For the evaluation, we used general evaluation metrics, precision [4] , recall [5], and the F-measure [6]. To measure the system's performance, we calculated the precision and the recall for the top N significant words of the baseline system and the re-evaluated system.

4.1 Estimate of the Size of Cut-off Term

In our experiments, we used the metrics of precision and recall to evaluate our method. These metrics are based on the number of synonymous expressions correctly extracted in the top *N* ranking. To define this cut-off term rank *N* for the data, we did some preliminary experiments with a small amount of data.

With the simple noise deletion approach we expect to increase the precision, however, the recall is not expected to be increased by using this method. We defined the maximum top value of *N* as satiation.

Figure 2 shows the performance against rank *N* for the entire corpus. We can see the satiation point at 20 in the figure. Therefore, we set *N* equal to 20 in our experiments for synonymous expression extraction from the entire corpus.

At the same time, we want to know the highest value of *n* to obtain the noise candidates. In each author's corpus a lower recall is acceptable, because we will remove these words as noise from the results of the entire corpus.

These results lead to the conclusion that the window size of the rank *N* for the entire corpus and the

[4] $Precision = \frac{Number\ of\ synonyms\ correctly\ extracted}{Number\ of\ synonyms\ extracted}$
[5] $Recall = \frac{Number\ of\ synonyms\ correctly\ extracted}{Number\ of\ synonyms\ in\ answer\ set}$
[6] $F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$

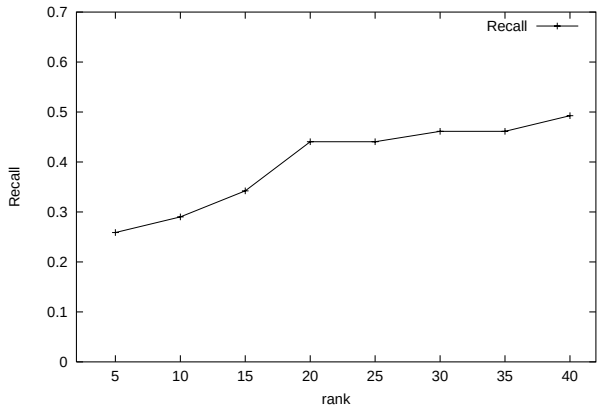

Figure 2: The Recalls of the Synonymous Extraction System Against the Rank

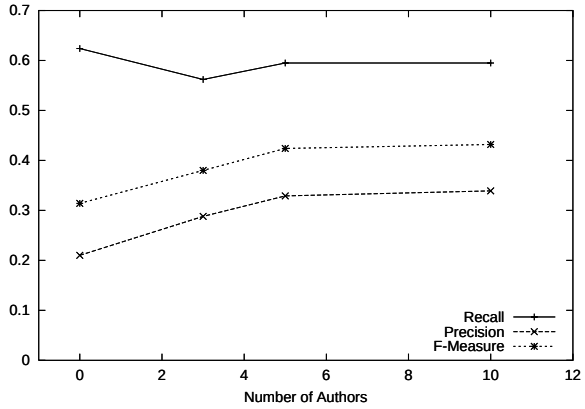

Figure 3: The Results After Noise Reduction by Using Authors' Corpora

rank *n* for each corpus should have the same value, *20*. During the evaluation, we extracted the synonymous expressions with the top *20* similarities from the entire corpus and removed the noise candidates with the top *20* similarities from each author's corpora.

4.2 Most Relevant Word Approach

The basic idea of this method is that one author mostly uses a unique expression to represent one meaning. According to this idea, the most similar words in each authors' corpora tend to be synonymous expression candidates. Comparing these two methods, one is a system for removing noise and the other is a system for extracting the most similar word.

According to the assumption of one person mostly using one unique expression to represent one meaning, we can extract the synonymous expressions that are the most similar word to the target word in each author's corpus. In comparison with the approach using the most similar word in each author's corpus and removing the noise, we calculated the recall rates for the most similar word approach. Table 6 shows the recall rates for the system with the entire corpus, the system using the top word from three authors' corpora, five authors' corpora, and ten authors' corpora.

	entire corpus	3 authors	5 authors	10 authors
Recall	0.624	0.114	0.114	0.143

Table 6: The Recall when Defining the Most Similar Words as Answers

These results show that the most similar words in the authors' corpora are not necessarily synonymous expressions for the target word, since some authors use other expressions in their corpus.

4.3 Noise Deletion Approach

For evaluating the deleting noise approach, the performance against the number of authors is shown in Figure 3. We extracted the top 20 synonymous expression candidates from the entire corpus, and removed the top 20 (except for the most similar words) noise candidates from the authors' corpora. Figure 3 contains the entire corpus result, and the results after removing the noise from three authors' corpora, five authors' corpora, and ten authors' corpora.

This figure shows that the noise reduction approach leads to better precision than the basic approach, but the recall rates are slightly reduced. This is because they sometimes remove words that are not noise, when an author used several expressions for the same word. In spite of that, the F-measures are increased, showing the method improves the accuracy by 37% (when using 10 authors' corpora). In addition, the table indicates that the improvement relative to the number of authors is not yet at a maximum.

5 Related Work

There have been many approachs to automatic detection of similar words from text. Our method is similar to (Hindle, 1990), (Lin, 1998), and (Gasperin, 2001) in the use of dependency relationships as the word features. Another approach used the words' distribution to cluster the words (Pereira, 1993), and Inoue (Inoue, 1991) also used the word distributional information in the Japanese-English word pairs to resolve the polysemous word problem.

Wu (Wu, 2003) shows one approach to collect synonymous collocation by using translation information. This time we considered only synonymous expression terms, but the phrasal synonymous expression should be the target of aggregation in text analysis.

Not only synonymous expressions, but abbreviation is one of the most important issues in term aggregation. Youngja (Youngja, 2001) proposed a method for finding abbreviations and their definitions, using the pattern-based rules which were generated automatically and/or manually.

To re-evaluate the baseline synonym extraction system, we used the authors' writing styles, and there are some researches using this approach. The most famous usage for them is the identification of a unknown author of a certain document (Thisted, 1987).

6 Conclusion and Future Work

This paper describes how to use the coherent corpus for term aggregation. In this paper we used the personal stylistic variations based on the idea that one person mostly uses one expression for one meaning. Although variations of personal writing styles are cause of the synonymous expressions in general, we managed to take advantage of such personal writing styles in order to reduce noise for term aggregation system.

We argued mainly about synonymous expressions in this paper, we can extract abbreviations and frequent missspelled words, and they should be considered as terms in term aggregation. We have to consider not only role-based word similarities, but also string-based similarities.

In general, a wide range of variations in expressions for the same meaning is a problematic feature of noisy data. However, in our method, we exploit these problematic variations for useful information for improving the accuracy of the system. This noise removal approach is effective when the data contains various expressions coming from various authors. Gasperin (Gasperin, 2001) indicated the specific prepositions are relevant to characterize the significant syntactic contexts used for the measurement of word similarity, considering what prepositions do and do not depend on personal writing style remains as future work.

In this paper, our work is based on the call center's logs, but this method is suitable for data from other domains. For example we anticipate that patent application data will be a suitable resource, because this data includes various expressions, and the expressions are based on each company's terminology. On the other hand, e-mail data does not seem suitable for our approach because other authors influence the expressions used. While we restricted ourselves in this work to this specific data, our future work will include an investigation of the character of the data and how it influences our method.

References

Charniak, E. 1993. *Statistical Language Learning*. MIT press.

Caroline Gasperin, Pablo Gamallo, Alexandre Agustini, Gabriel Lopes, and Vera de Lima 2001. Using Syntactic Contexts for Measuring Word Similarity *In the Workshop on Semantic Knowledge Acquisition & Categorisation (ESSLLI 2001)*

Donald Hindle 1990. Noun Classification From Predicate-Argument Structures. *Proceedings of the 28th Annual Meeting of ACL*, pp.268-275

Naomi Inoue 1991. Automatic Noun Classification by Using Japanese-English Word Pairs. *Proceedings of the 29th Annual Meeting of ACL*, pp. 201-208

Dekang Lin 1998. Automatic Retrieval and Clustering of Similar Words *COLING - ACL*, pp768-774,

Nasukawa T. and Nagano, T. 2001. Text analysis and knowledge mining system. In *IBM Systems Journal*, Vol. 40, No. 4, pp. 967–984.

Mary S. Neff, Roy J. Byrd, and Branimir K. Boguraev. 2003. The Talent System: TEXTRACT Architecture and Data Model. In *Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology systems (SEALTS)*, pp. 1–8.

Youngja Park and Roy J. Byrd 2001. Hybrid text mining for finding abbreviations and their definitions. *Proceedings of the 2001 Conference on EMNLP*, pp.126-133

Fernando Pereira and Naftali Tishby 1993. Distributional Clustering of English Words *Proceedings of the 31th Annual Meeting of ACL*, pp. 183-190

Strzalkowski T. and Vauthey B. 1992. Information Retrieval Using Robust Natural Language Processing. *Proceedings of ACL-92*, pp.104-111.

B. Thisted and R. Efron. 1987. Did Shakespeare write a newly discovered poem?. *Biometrika*, pp. 445–455

Hua Wu and Ming Zhou 2003. Synonymous Collocation Extraction Using Translation Information *Proceedings of the 41st Annual Meeting of ACL*, pp.120-127