Automatic Expansion of Abbreviations by using Context and Character Information [⋆]

Akira Terada [*], Takenobu Tokunaga, Hozumi Tanaka

Department of Computer Science,
Tokyo Institute of Technology
2-12-1 Ôokayama Meguro, Tokyo, 152-8552, Japan

Abstract

Unknown words such as proper nouns, abbreviations, and acronyms are a major obstacle in text processing. Abbreviations, in particular, are difficult to read/process because they are often domain-specific. In this paper, we propose a method for automatic expansion of abbreviations by using context and character information. In previous studies dictionaries were used to search for abbreviation expansion candidates (candidates words for original form of abbreviations) to expand abbreviations. We use a corpus with few abbreviations from the same field instead of a dictionary. We calculate the adequacy of abbreviation expansion candidates based on the similarity between the context of the target abbreviation and that of its expansion candidate. The similarity is calculated using a vector space model in which each vector element consists of words surrounding the target abbreviation and those of its expansion candidate. Experiments using approximately 10,000 documents in the field of aviation showed that the accuracy of the proposed method is 10% higher than that of previously developed methods.

Key words: Abbreviation Expansion, Context Information, Unknown Words

1 Introduction

The presence of unknown words degrades the performance of text processing applications, such as information retrieval and text data mining. By unknown words, we mean words

[⋆] This is a revised version paper appeared in the *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium Workshop on Automatic Paraphrasing:Theories and Applications (Terada & Tokunaga, 2001).*

[*] Corresponding author. Tel.: +81-3-5734-2831; fax: +81-3-5734-2915
Email address: aterada@cl.cs.titech.ac.jp (Akira Terada).

and symbols that are not recognized by the system, including numbers, proper nouns, abbreviations, acronyms, misspelled and run-on words, and so on. In this paper, we focus on abbreviations, and propose a method for expanding them into original words.

Abbreviations are a short representation of words, and they are used in newspaper headlines, captions, and other types of text to save space and give readers an impression of what abbreviations mean. Although the use of abbreviations does save space, they are sometimes hard to understand by those who are not familiar with the field where these abbreviations are commonly used. Readers can expand abbreviations by looking at words around them (context information) and by analyzing the characters in the abbreviations. We can also learn the meaning of abbreviations from the text in which original word forms are used. We simulated this " human " way of expanding abbreviations on a computer. For context information, we used words appearing before and after abbreviations and abbreviation expansion candidates. To calculate context information, we used cosine similarity which is known to be a good metric. For characters information, we explore the human tendency of abbreviating words and make rules.

We define an abbreviation as a short representation of a single word, e.g., "TKOF - takeoff", and we define an acronym as a short representation of more than one word, e.g., "ILS - Instrument Landing System". Hereafter, we show abbreviations by using upper case letters, and their original word forms by using lower case letters.

We made the following assumptions about abbreviations:

(1) Abbreviations include fewer characters than their original word forms.
(2) The characters used in an abbreviation are a subset of those appearing in the original word form in the same order (e.g., "TKOF – takeoff"), except for "X", "–", and "/" (e.g., "WX – weather").
(3) Abbreviations are not real words.
Some abbreviations are registered in some dictionaries, but not in others. We define an abbreviation as a word which is not registered in our system dictionary.
(4) Abbreviations correspond to unique words.
In general, assumption (4) does not hold, but if we restrict the application to a specific domain, most abbreviations correspond to unique words.

Previous abbreviation expansion methods returned many abbreviation candidates because of their low precision, forcing the user to make the final choice (Rowe. & Laitinen., 1995). When used for such tasks as text-to-speech synthesis, however, the system must perform fully automatic abbreviation expansion.

Most existing abbreviation expansion methods use only the abbreviations themselves, without taking into account any information that can be gleaned from the linguistic or textual context in which the abbreviations appear (Rowe. & Laitinen., 1995) (Uthurusamy., Means., & Godden, 1993) . When users expand an abbreviation into its original word, they rely heavily on context information around the abbreviation (Rowe. & Laitinen., 1995). To improve the system performance, we need context information.

One may think that if there is a fixed list of abbreviations, there is no problem. However, in the real world, different people or even the same person may abbreviate the same word differently, so a fixed list of abbreviations has limited applications (e.g., both “A/C” and “ACFT” stand for “aircraft”). Also the use of such a fixed list of abbreviations requires some effort on the part of the user to look up an abbreviation list and on the part of administrators to maintain such a list.

The rest of this paper is organized as follows. In Section 2, we describe abbreviations frequently used in the field of aviation. In Section 3, we review related studies on abbreviation expansion. Section 4 describes the design of our system and various methods of abbreviation expansion. Section 5 presents experimental results of our system and compares these results with those obtained by using other methods. Section 6 discusses potential problems with our approach. Section 7 shows our conclusion and outlines future work.

2 Target Domain

The field of aviation involves frequent use of abbreviations. In this study, we used a data set from the ASRS (Aviation Safety Reporting System) database[1], where abbreviations make up approximately 11.0% and acronyms make up 2.0% of all words based on a manual count of words in an 8,000-word sample. In general, abbreviations/acronyms are not so frequent. In 145,295-word documents excerpted from 1989 Wall Street Journal, we found only 0.003% of abbreviations and 0.01% of acronyms, excluding the words which were registered in the system, such as “Nov” and “USA”. However less formal texts include more abbreviations/acronyms. The ASRS database is a collection of reports submitted by pilots, air traffic controllers, flight attendants, mechanics, ground personnel, and others who observed or were involved in aviation incidents. Incident reports are read by at least two ASRS aviation safety analysts. This report system is designed to preserve the confidentiality and anonymity of the reporter, so all information that may be used to identify the reporter including the name of the aircraft is deleted from the report. Thus unspecified names such as “X”, as in “aircraft X” are often used. These unspecified names are also unknown words. Our task is to distinguish these unspecified names from abbreviations. The ASRS data are structured and include the report number, local date, weather conditions, and the text of each report, which is further divided into a narrative part and a synopsis. The synopsis is shorter than the narrative and is a summary of the narrative. We used the narratives from the ASRS dataset, which averaged 38 words per document in our experiment.

To search for abbreviation expansion candidates, we used a corpus from the same field with few abbreviations that we called “ an abbreviation-poor corpus ”. An abbreviation-rich corpus includes many abbreviations and an abbreviation-poor corpus does not include many abbreviations. To rank abbreviations effectively and efficiently, we propose a method

[1] ASRS data are available at http://nasdac.faa.gov/asp/asy_asrs.asp

of abbreviation expansion that mainly uses context information, i.e., words appearing before and after abbreviations. Using this abbreviation-poor corpus, we collected words surrounding abbreviation expansion candidates. We assumed that the words appearing near an abbreviation and those appearing near its original word form are statistically similar. The idea is that if we use an abbreviation-poor corpus, we can obtain abbreviation expansion candidates appearing in an abbreviation-rich corpus, without using a dictionary. There are two reasons why we used an abbreviation-poor corpus: (1) such a corpus contains information on word frequency, and (2) an abbreviation-rich corpus may not include the original form of a given abbreviation. For an abbreviation-rich and an abbreviation-poor corpus from the same field, we can expect that most of original word forms of abbreviations in the abbreviation-rich corpus will be included in the abbreviation-poor corpus with similar context.

For this purpose, we used a data set from the NTSB (National Transportation Safety Board)[2] database containing synopses of aviation accidents as our abbreviation-poor corpus. In our ASRS dataset, "TKOFF" appears 1,156 times, while "takeoff" does not appear even once ("TKOFF" is an abbreviation of "takeoff" in the ASRS database). In contrast, in the NTSB dataset, "TKOFF" appears 28 times and "takeoff" appears 344 times . The statistics of the ASRS and NTSB datasets are given in Table 1.

Table 1
Statistics of the ASRS and NTSB datasets

	ASRS	NTSB
No. of documents	2,648	3,937
No. of words (tokens)	677,725	426,717
No. of words (types)	18,422	14,940
Ave. no. of words/ doc.	260	108
No. of unknown words (tokens)	100,983	33,575
No. of unknown words (types)	6,077	7,190
Ave. no. of unknown words/ doc.	38.1	8.5
Unknown word ratio	14.9%	8.5%

3 Related Work

There have been several attempts to expand abbreviations by searching their definitions in the local context (Park & Byrd, 2001) (Larkey, Ogilvie, Price, & Tamilio, 2000) and they actually achieved a good performance. These attempts assume that abbreviations and acronyms are introduced with their definition when they are used in the first time.

[2] URL:http://www.ntsb.gov/aviation/aviation.htm

This assumption could be hold in texts of a general domain, but not always does in specific domains such as aviation reports, maintenance logs and so on. In this paper we do not assume that the definitions of abbreviations and acronyms appear in their local context.

On the surface, abbreviation expansion appears to be similar to spelling error correction (Kukich, 1992b), but as Rowe et al. pointed out, abbreviations are much shorter than their original word forms and contain less information than misspelled words (Rowe. & Laitinen., 1995). Kukich reported that most non-word errors tend to occur within two characters of the correct spelling (Kukich, 1992b). In our experiment, we found that abbreviations are 4.3 characters shorter than their original word forms. This means that it is difficult to extract the original form of an abbreviation by using only character information.

However, most existing abbreviation expansion systems use mainly character information (Uthurusamy. et al., 1993) (Toole, 2000). Uthurusamy et al. performed abbreviation expansion by using different algorithms to rank abbreviations candidates by using contraction-type and truncation-type information after detecting unknown words, correcting spelling errors, and choosing abbreviation candidates in a text-correction and standardization system (Uthurusamy. et al., 1993). Rowe et al. also ranked abbreviation candidates by using rules, and he used only character information for context information. Toole divided abbreviation expansion tasks into two steps: detection and expansion. Toole used a binary weight of context (for both detection and expansion). If a candidate word occurs in the same context, it is assigned a score of one, otherwise it is zero. For example, if the abbreviations “ALT” is considered and “ALT” occurs in the context of “crusing ALT was”,and the candidate word “altitude” appears in the context of “cruising altitude” or “altitude was”, then the contextual weight is one. If altitude does not occurr in the corpus with either of these words, then weight is zero. They concluded these features are not predictive (Toole, 2000). We think one reason why Toole’s contextual information was not predictive is that Toole used only one window size and exact match of the word appearing before and after abbreviations.

An abbreviation expansion task can be divided into three sub-tasks: detection, expansion, and ranking. Abbreviation detection is the easiest of these sub-tasks. Rowe et al. and Toole used a dictionary to search for abbreviation expansion candidates (Rowe. & Laitinen., 1995; Toole, 2000). In their method, because a general dictionary was used, many irrelevant abbreviation expansion candidates were produced, which degraded the system performance.

4 System Design

We consider unknown words to be abbreviation candidates (assumption (3) in Section 1). To detect unknown words, we take a tagged corpus and use “unknown word” tags to annotate unknown words by using a tagger. When an unknown word appears, it is an abbreviation candidate, but it may or may not be an abbreviation. For every unknown word, the system searches for abbreviation expansion candidates from an abbreviation-poor corpus using assumptions (1) and (2). Then the system collects the words around the abbreviation candidates and abbreviation expansion candidates for context information. Using this information, the system determines which abbreviation expansion candidate is the most appropriate for the original form of the target abbreviation, or it may determine that none of the abbreviation expansion candidates is appropriate for the original form of the target abbreviation, that is, the system decides that the unknown word is not an abbreviation.

Our method can be divided in three steps : abbreviation candidate detection, abbreviation expansion candidate detection, and ranking.

We tagged the ASRS and NTSB data by using TreeTagger (Schmid, 1994, 1995). TreeTagger annotates text with part-of-speech and lemma information. TreeTagger annotates words as “unknown” when they are not listed in the tagger dictionary. Because abbreviations are not real words, they are annotated as “unknown”. However, even if the tagger annotates words as “unknown”, it guesses their part of speech and assigns corresponding tags to them along with the “unknown” tag.

First, to detect an abbreviation candidate, we take “unknown” words as abbreviation candidates only when they are guessed to be nouns, verbs, adjectives, adverbs, or prepositions (POS filter).

For context information, words appearing before and after an abbreviation candidate are collected. Because TreeTagger provides the root form of each word, we use this information in word collection, and no further stemming is necessary. In collecting words around abbreviation candidates, all abbreviations of the same type are collected (assumption (4)). We used a window of size 3. In our preliminary test, we experimented with windows of sizes 3 and 5. We found no noticeable difference in the results. The window size may be greater than the syntactical scope of a word, but this is beyond the scope of this paper.

Second, we searched for abbreviation expansion candidates of abbreviations in the abbreviation-poor corpus by using assumptions (1) and (2): Abbreviation expansion candidates include more characters than abbreviation candidates and abbreviation expansion candidates include the same characters in the same order as abbreviation candidates except for “X”, “-”, and “/”. For example, if an abbreviation candidate is “FLT”, its abbreviation expansion candidates are “flight”, “filters”, “flat”, “float”, “difficult”, “flights”, “felt”, and “floated”. We take only nouns, verbs, adjectives, adverbs, and prepositions as abbreviation expansion candidates (POS filter). A window of the same size was used here to collect words around abbreviation expansion candidates as in collecting abbreviation expansion context information.

Finally, abbreviation expansion candidates are filtered by using rules (we will discuss this in detail in section 4.3) and ranked by using context information.

Figure 1 illustrates how our system works , and Figure 2 shows an example of context information.

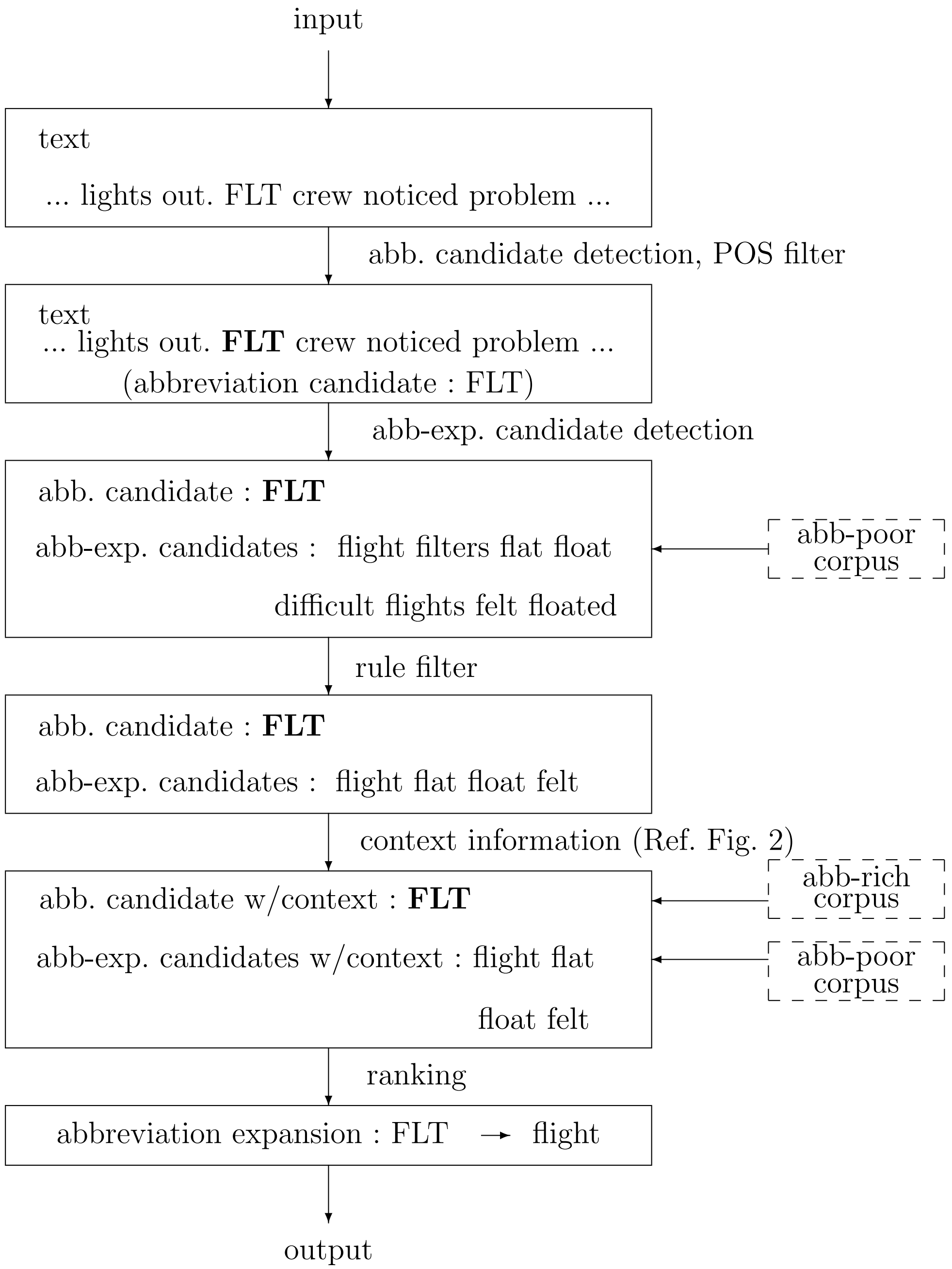
input
text
... lights out. FLT crew noticed problem ...
abb. candidate detection, POS filter
text
... lights out. FLT crew noticed problem ...
(abbreviation candidate : FLT)
abb-exp. candidate detection
abb. candidate : FLT
abb-exp. candidates : flight filters flat float
difficult flights felt floated
abb-poor corpus
rule filter
abb. candidate : FLT
abb-exp. candidates : flight flat float felt
context information (Ref. Fig. 2)
abb. candidate w/context : FLT
abb-exp. candidates w/context : flight flat
float felt
abb-rich corpus
abb-poor corpus
ranking
abbreviation expansion : FLT → flight
output

Fig. 1. System Overview

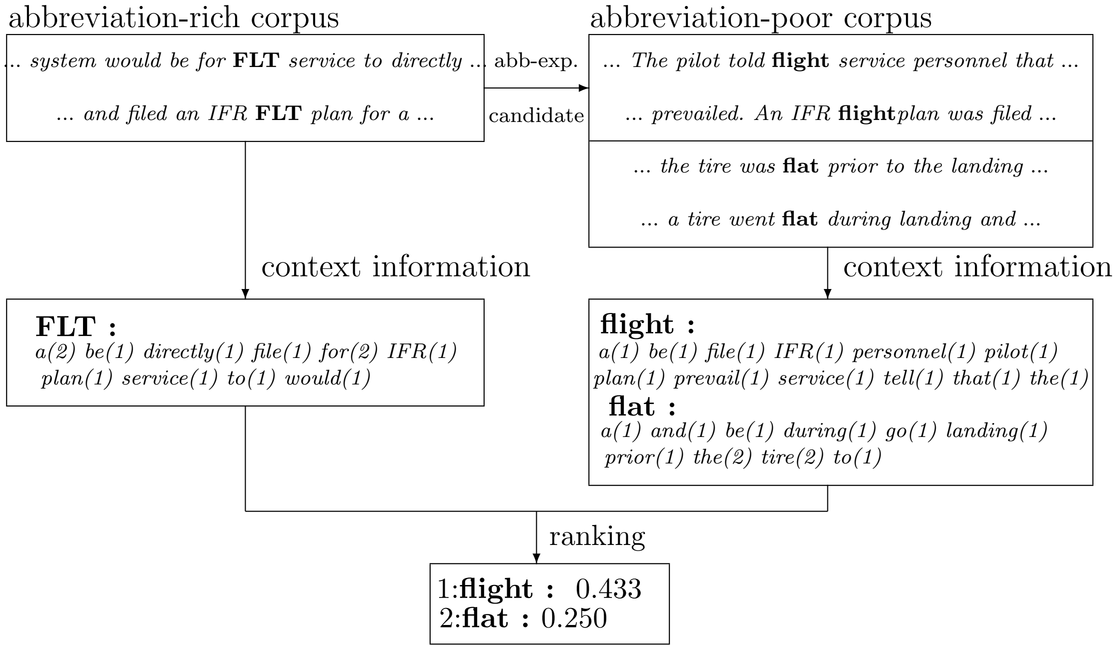
abbreviation-rich corpus
... system would be for FLT service to directly ...
... and filed an IFR FLT plan for a ...
abb-exp.
candidate
abbreviation-poor corpus
... The pilot told flight service personnel that ...
... prevailed. An IFR flightplan was filed ...
... the tire was flat prior to the landing ...
... a tire went flat during landing and ...
context information
context information
FLT :
a(2) be(1) directly(1) file(1) for(2) IFR(1) plan(1) service(1) to(1) would(1)
flight :
a(1) be(1) file(1) IFR(1) personnel(1) pilot(1) plan(1) prevail(1) service(1) tell(1) that(1) the(1)
flat :
a(1) and(1) be(1) during(1) go(1) landing(1) prior(1) the(2) tire(2) to(1)
ranking
1:flight : 0.433
2:flat : 0.250

Fig. 2. An example of context information

4.1 Term Weighting

We take the word context around an abbreviation to be a query and each expansion candidate to be a document and the word context around an expansion candidate to be the contents of a document, in information retrieval terms. In information retrieval, many of the most frequently occurring words such as “the” and “or” are eliminated from indexing terms, because these words have no specific meaning in each individual document (Frakes & Baeza-Yates, 1992). However, in our case, many abbreviations are surrounded by function words representing the nature of abbreviations, because an abbreviation is not a document, but a single word. We believe that words such as “CLRD – cleared” are often followed by words like “to” or “for” (for example, “cleared to land”, “cleared for takeoff”), and that eliminating these words will create an adverse effect. Our experimental results supported this design. Thus, we did not eliminate function words.

Term weights are assigned by using either *tf* (term frequency) or *tf-idf* (term frequency-inverse document frequency) in both documents and queries. In information retrieval, *tf* contributes to recall and *idf* is used to improve precision. *tf* and *idf* are usually combined by multiplying *tf* by *idf*, which is denoted by *tf-idf*. In weighting the words, we evaluated our system by using both *tf* and *tf-idf* in our preliminary test, and found no noticeable difference between them. We thus chose to use *tf* for weighting. Hearst also posited that *tf-idf* is not accurate when used for comparing adjacent pieces of a text, and *tf* appears to be more robust for this purpose (Hearst, 1997). We think that using *idf* may degrade the system performance because rare words sometimes found around abbreviations or abbreviation expansion candidates may not reflect the nature of these abbreviations. Also, viewing abbreviations as types and collecting all words around a given type may cancel out rare words.

4.2 Ranking

To rank candidate words, a vector space model (Salton, 1988) is used, in which documents and queries are represented in a multi-dimensional space. Each dimension corresponds to a document/query term.

To measure the vector similarity, we used the following cosine metric:

$$score = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

where x_i and y_i are the weights of the words appearing around an abbreviation and an abbreviation expansion candidate, respectively.

Kukich reported that the cosine metric can best be used for spelling correction in terms of its overall effectiveness and efficiency (Kukich, 1992a).

4.3 System Decision

The system must choose the final answer from abbreviation expansion candidates, or it must determine that none of the abbreviation expansion candidates is an abbreviation. To do this, the system uses various methods to filter out abbreviation expansion candidates and return the highest scoring abbreviation expansion candidates as the answer.

(1) Method 1
The highest scoring abbreviation expansion candidate is the original word form of the target abbreviation.

(2) Method 2
As Uthurusamy et al. pointed out, English speakers tend to truncate (cut off a word-final sub-string) or contract (remove internal characters) words when abbreviating them (Uthurusamy. et al., 1993). We make use of these properties by introducing a C4.5 (Quinlan, 1993) decision-tree-based classification system with the eight features described below. First the system finds abbreviation expansion candidates which satisfy the decision tree. Then the system choose the highest scoring abbreviation expansion candidate which is classified as an "abbreviation" by the decision tree.

- First character matching:
  Abbreviations usually begin with the same character as the original word form, except when they begin with an "X", e.g., "XING – crossing".

- Final character matching:
  In many cases, an abbreviation and its original word form have the same final characters. This feature means the number of final characters which are shared between the abbreviation and the original word.
- Truncation:
  e.g., “CAPT – captain”
- Missing vowels:
  A lack of vowels in abbreviations compared to abbreviation expansion candidates provides a strong indication of an abbreviation expansion candidate,
  e.g., “DSCNT – descent”
- Single sub-string with vowels and consonants:
  An abbreviation derived from its original word form by removing a single sub-string including both vowels and consonants provides a strong indication of abbreviation expansion candidate, e.g., “CLB – climb”
- Single sub-string with consonants:
  An abbreviation derived from its original word by removing a single sub-string including only consonants provides a strong indication an inappropriate abbreviation expansion candidate,
  e.g., “CLIB – climb”
- ASRS frequency:
  The frequency of occurrence of an abbreviation in the ASRS dataset.
- NTSB frequency:
  The frequency of occurrence of an abbreviation expansion candidate in the NTSB dataset.

Besides the above eight features, the unknown-word length is a good criterion for differentiating between abbreviations and real words. In our system, however, long unknown words (e.g., proper nouns or run-on words) were mostly rejected because they had no candidate words in the abbreviation-poor corpus. So we did not use this feature.

(3) Method 3
We replaced all initial Xs with “TRANS”, “CROSS” and “OUT” to find candidate words before using method 2.

(4) Method 4
In the aviation field, acronyms too are used frequently. However, acronyms are used consistently in every text, e.g., “GS – glide slope”. Before using method 3, we used a fixed acronym list containing 114 acronyms to distinguish between abbreviations and acronyms.

(5) Method 5
In the aviation field, IATA (International Air Transport Association) airport 3-letter codes are commonly used. Before using method 4, we used IATA airport codes to distinguish between abbreviations and airport codes.

5 Evaluation

We examined 10,000 ASRS documents in our experiment. In these ASRS documents, 2,648 documents included narratives, so we used these 2,648 documents (Table 1). In these documents, 265 different abbreviation types were found by manually checking all unknown words detected by the system. However, other abbreviations were also included in the ASRS dataset that were not identified as " unknown words " by the tagger.

From the 2,648 documents, we chose 20 documents from which we extracted different abbreviation types for the test. We thus obtained 106 different abbreviation types. Among these 106 abbreviation types, 84 were detected as abbreviation candidates by the system, and 22 were not detected as abbreviation candidates, because the tagger did not identify them as unknown words. Of the 84 abbreviation candidates, four abbreviations did not have correct abbreviation expansion candidates in the NTSB dataset (two had no abbreviation expansion candidates and the other two had no correct abbreviation expansion candidates). The abbreviation "MI" corresponded to both "miles" and "minutes" in the 20 document samples (assumption (4) did not hold). Thus we had a total of 107 different abbreviations in the documents. Because the system did not detect 22 of these as unknown words, these 22 words were not included in the 265 different abbreviation types that the system detected. We used the remaining 181 abbreviations types for training (181=265-(106-22)). In this way, we avoided including the same abbreviation types in both the training and test data.

To evaluate the system, we used precision and recall. Precision and recall are defined as follows:

$$precision = \frac{w}{w+x+y}$$

$$recall = \frac{w}{w+x+z}$$

where w is the number of abbreviation types expnadeded correctly; x is the number of abbreviation types expanded incorrectly[3]; y is the number of abbreviation candidate types that were not abbreviations but were incorrectly expanded by the system; and z is number of abbreviation types not detected as abbreviations by the system.

To evaluate our system, we used the methods described in section 4.3. When we used method 5, recall was low because there was inconsistency in the use of abbreviations (assumption (3) did not hold), e.g., "SVC" stood for both "service" and an airport name. Table 2 shows the results for the 20 documents.

To avoid tagger errors, we manually labeled the 22 undetected words as "unknown words".

[3] The first author, who has 20 years of experience working in the aviation field, made an answer list.

Table 2
Results of 20 documents

	precision (%)	recall (%)
Method 1	40.4 (55/136)	51.4 (55/107)
Method 2	63.3 (57/90)	53.3 (57/107)
Method 3	64.4 (58/90)	54.2 (58/107)
Method 4	70.7 (58/82)	54.2 (58/107)
Method 5	81.4 (48/59)	44.9 (48/107)

The results in Table 3 show that precision was almost the same as in the results in Table 2 , but recall was about 10% higher. Thus if a tagger dictionary is constructed for a specific field, recall can be improved without degrading precision.

Table 3
Results for 20 documents after re-labeling 22 undetected abbreviations as unknown words

	precision (%)	recall (%)
Method 1	40.5 (64/158)	59.8 (64/107)
Method 2	60.9 (67/110)	62.6 (67/107)
Method 3	61.8 (68/110)	63.4 (68/107)
Method 4	66.3 (67/101)	62.6 (67/107)
Method 5	73.2 (52/71)	48.6 (52/107)

To evaluate the system performance for the whole document set, we defined recall* as shown below, because we could not check all the data for abbreviations that were not recognized as " unknown words " by the tagger.

$$recall* = \frac{w}{w+x}$$

As mentioned above, if a tagger dictionary is constructed for a given field, " recall* " is equal to " recall ". Table 4 shows these results.

Next, we compared our results with the results of some previous studies by using precision, recall, and F-measure. F-measure is a combination of precision and recall into a single measure of overall performance (van Rijsbergen C.J., 1979) in our experiment with equal weight of precision and recall.

$$F = \frac{2PR}{P+R}$$

Table 4
Results for the whole document set

	precision (%)	recall* (%)
Method 1	23.9 (102/426)	38.3 (102/266)
Method 2	51.9 (110/212)	41.4 (110/266)
Method 3	54.1 (111/205)	41.7 (111/266)
Method 4	58.1 (111/191)	41.7 (111/266)
Method 5	70.7 (94/133)	35.3 (94/266)

where F is F-measure, P is precision, and R is recall. Toole used ASRS dataset and the test set was the same domain as ours, but the test set was not the exactly the same as ours, so we cannot directly compare our results with hers. Toole performed a two-step disabbreviation task, i.e., identification and expansion, by using an ASRS dataset. Toole reported that 188 (different) unknown words were detected in the abbreviation identification step. Of the 188 unknown words, some were not abbreviations, but they were incorrectly identified as such in the identification process. Toole reported that there were only a few errors. Recall of 58.5% was obtained when the abbreviation expansion process returned only one abbreviation expansion candidate. Toole did not evaluate precision, but we estimate that precision was several percent lower than recall, because there were a few mistakes in the abbreviation identification step. (Toole, 2000). Our system's recall was 4-5% lower, but precision was 6-12% higher than in the results of Toole. Our system's F-measure was about 0-3% higher than in the results of Toole (Table 5).

Table 5
Comparison with previous results (1)

	precision (%)	recall (%)	F (%)
(Toole 2000)	< 58.5	58.5	< 58.5
Ours (method 3)	64.4	54.2	58.9
Ours (method 4)	70.7	54.2	61.4
with tagger error correction			
Ours (method 3)	61.8	63.4	62.6
Ours (method 4)	66.3	62.6	64.4

Rowe et.al proposed a system that works interactively with the user. That is, the system presents to the user up to five abbreviation expansion candidates one by one. The user determines whether a candidate is the original word form of the abbreviation in question. Rowe et.al performed an abbreviation expansion task by using photograph captions, and designed a system that accepts phrases with up to two words (Rowe. & Laitinen., 1995).

Toole recalculated her results to make them comparable to (Rowe. & Laitinen., 1995)and we also recalculated our results in the same way (Toole, 2000). Our system recall was about 10% lower than that of the previous studies, but precision was 15-60% higher. Our system's F-measure was 10-45% higher than previous two studies (Table 6).

Table 6
Comparison with previous studies/results (2)

	precision (%)	recall (%)	F (%)
(Rowe et al. 1995)	14.9	71.1	24.6
(Toole 2000)	32.8	69.6	44.6
Ours (method 3)	49.2	59.8	54.0
Ours (method 4)	76.8	58.9	66.7
with tagger error correction			
Ours (method 3)	46.9	71.0	56.5
Ours (method 4)	73.1	69.2	71.1

We believe that if a tagger dictionary specifically created for this field had been used, recall would have been approximately the same as in Toole's results (Table 6 lower part).

We compared our results with those obtained by human annotators. We generated evaluation data for 20 documents which were same as test data previously used for three annotators who were familiar with the aviation field. The annotators' precision ranged from 85.7% to 94.9%, and their recall ranged from 50.4% to 78.5%. F-measure ranged from 65.8% to 85.5%, and the average F-measure was 77.7%. These results show that our system's F-measure obtained using method 4 (which shows the best result) is about 4% lower than the lowest F-measure of the human annotators and 16% lower than the average F-measure of the human annotators. Thus, our system's performance is somewhat lower than that of human annotators familiar with the field, however, our system should prove itself useful on the large scale text.

6 Discussion

We conducted an abbreviation expansion experiment using an ASRS dataset, in which abbreviations were included in approximately 11.0% of all entries. Newspapers and magazines do not usually contain so many abbreviations. However, the number of abbreviations in automobile maintenance records or aircraft maintenance logbooks is comparable to that in the dataset we used. So we think our methods may be used in other fields.

We used an abbreviation-poor corpus instead of a dictionary to search for abbreviation expansion candidates. One may think that an abbreviation-poor corpus is not always available. In this case, for a reporting system like ASRS, we can create an abbreviation-poor corpus by asking analysts not to use abbreviations for a certain period. An abbreviation-poor corpus can be considered a training corpus. If a corpus includes both abbreviations and their original forms, we may use this corpus both as an abbreviation-rich and an abbreviation-poor corpus. Our NTSB dataset is one example of such a corpus, and we plan to test this idea.

The results of method 4 show that 24 out of 82 answers were incorrect. Of these, five were acronyms used in the ASRS dataset, five were airport code names, one was due to a word not registered in the tagger dictionary (RETUNED), and 13 were abbreviations detected as abbreviations, but expanded wrongly. Of these errors, two were inflection errors, e.g., "DEG" was expanded to "degrees", while "DEGS" was correctly expanded to "degrees". Such errors can be corrected by finding the stem of the target word (in this case, "DEG" – "degree") and expanding its inflection forms.

Among the abbreviations not detected as abbreviations by the system, four had no abbreviation expansion candidates in the abbreviation-poor corpus. Another two had abbreviation expansion candidates, but the frequencies of their occurrence were low, so these two abbreviations were not detected as abbreviations by the system.

To determine whether using context information is effective in ranking abbreviations, we compared the results obtained by using method 3 with and without context information (Table 7). When context information was used, the system's F-measure was improved by about 23%.

Table 7
Results obtained using method 3 with and without context information

	precision (%)	recall (%)	F (%)
method 3 with context information	64.4	54.2	58.9
method 3 without context information	25.9	59.8	36.1

The tag information of unknown words was not useful, because many unknown words were tagged as proper nouns. So we could not use this feature for our system.

We manually labeled the 22 undetected words (AFT, ALT, APPROX, CA, COM, COMP, CRS, DES, INFO, LAV, MI, MIN, MINS, N, NE, NW, PAX, POS, REF, S, VIS, and W) as "unknown words" in Table 3. One may think that some abbreviations are listed in general-use dictionaries. We agree with this, but we wanted to check our system's performance in a conservative way. Some of the 22 undetected words are listed in some dictionaries (but not all) while others are not.

In methods 4 and 5, we used a domain-specific dictionary. In real applications, we can use a domain-specific dictionary to eliminate words in the dictionary from abbreviation candidate words before using methods 1 to 3.

7 Conclusion

In this paper, we proposed a method for abbreviation expansion using context information and obtained some encouraging results. For example, “S” was successfully expanded to “south”, a word chosen from among 2,725 abbreviation expansion candidates. This could not be achieved without context information, because “S” is just one letter, and it has minimal character information.

We used an abbreviation-poor corpus instead of a dictionary to search for abbreviation expansion candidates. By using this technique, our system automatically obtained domain-specific information. The abbreviation-poor corpus used in our experiment included about 7,700 words types, whereas Toole used a dictionary with about 35,000 entries (Toole, 2000) and Rowe et.al used one with about 29,000 entries (Rowe. & Laitinen., 1995). Our dictionary contained about 22–27% of the entries used in previous studies. This means that we can achieve high precision and recall by using our method.

In terms of portability, our method primarily uses context information, and we use only general characteristics of English in methods 1 through 3, so our method can easily be used in other fields. Without domain-specific knowledge, we can achieve high precision and recall. For example, “katakana” is frequently used in Japanese to write imported words. The same “katakana” words sometimes have different forms

. Our system can be used to decipher different forms of “katakana” words.

The problem of abbreviation inconsistency (the same abbreviation corresponds to different words) can be solved by using context information for each abbreviation token (not abbreviation type), but we have not experimented with this idea yet.

8 Acknowledgment

The authors would like to thank the anonymous referees for their invaluable comments on the earlier version of this paper. We also thank the personnel of NASA, FAA, and NTSB for allowing us to use aviation incident reports and accident analyses. We also thank annotators N. Murayama, T. Tsutumi, and Y. Nozawa of Japan Air Lines Co., Ltd., for their assistance.

References

Frakes, W. B., & Baeza-Yates, R. (1992). *Information Retrieval.* Prentice Hall.

Hearst, M. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, *23*, 33–64.

Kukich, K. (1992a). Spelling correction for the telecommunications network for the deaf. *COMMUNICATIONS OF THE ACM*, *35*(5), 80–90.

Kukich, K. (1992b). Techniques for automatically correcting words in text. *ACM Computing Surveys*, *24*(4), 377–439.

Larkey, L. S., Ogilvie, P., Price, M. A., & Tamilio, B. (2000). Acrophile: An automated acronym extractor and server. In *Proceedings of the ACM Digital Libraries conference*, pp. 205–214.

Park, Y., & Byrd, R. J. (2001). Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of EMNP2001.*

Quinlan, J. R. (1993). *C4.5:PROGRAMS FOR MACHINE LEARNING.* Morgan Kaufmann Publishers.

Rowe., N. C., & Laitinen., K. (1995). Semiautomatic disabbreviation of technical text. *Information Processing & Management*, *31*(6), 851–857.

Salton, G. (1988). *Automatic Text Processing.* Addison-Wesley.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pp. 44–49.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to German. In *EAL SIGDAT workshop.*

Terada, A., & Tokunaga, T. (2001). Automatic disabbreviation by using context information. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium Workshop on Automatic Paraphrasing:Theories and Applications*, pp. 21–28.

Toole, J. (2000). A hybrid approach to the identification and expansion of abbreviations. In *Proceedings of RIAO'2000*, Vol. 1, pp. 725–736.

Uthurusamy., R., Means., L. G., & Godden, K. S. (1993). Extracting knowledge from diagnostic databases. *IEEE Expert*, *8(6)*, 27–38.

van Rijsbergen C.J. (1979). *Information Retrieval.* Butterworths.