Automating Help-desk Responses: A Comparative Study of Information-gathering Approaches

Yuval Marom and Ingrid Zukerman
Faculty of Information Technology
Monash University
Clayton, VICTORIA 3800, AUSTRALIA
{yuvalm,ingrid}@csse.monash.edu.au

Abstract

We present a comparative study of corpus-based methods for the automatic synthesis of email responses to help-desk requests. Our methods were developed by considering two operational dimensions: (1) information-gathering technique, and (2) granularity of the information. In particular, we investigate two techniques – retrieval and prediction – applied to information represented at two levels of granularity – sentence-level and document level. We also developed a hybrid method that combines prediction with retrieval. Our results show that the different approaches are applicable in different situations, addressing a combined 72% of the requests with either complete or partial responses.

1 Introduction

Email inquiries sent to help desks often "revolve around a small set of common questions and issues".[1] This means that help-desk operators spend most of their time dealing with problems that have been previously addressed. Further, a significant proportion of help-desk responses contain a low level of technical content, corresponding, for example, to inquires addressed to the wrong group, or insufficient detail provided by the customer about his or her problem. Organizations and clients would benefit if the efforts of human operators were focused on difficult, atypical problems, and an automated process was employed to deal with the easier problems.

In this paper, we report on our experiments with corpus-based approaches to the automation of help-desk responses. Our study is based on a corpus of 30,000 email dialogues between users and help-desk operators at Hewlett-Packard. These dialogues deal with a variety of user requests, which include requests for technical assistance, inquiries about products, and queries about how to return faulty products or parts.

In order to restrict the scope of our study, we considered two-turn short dialogues, comprising a request followed by an answer, where the answer has at most 15 lines. This yields a sub-corpus of 6659 dialogues. As a first step, we have automatically clustered the corpus according to the subject line of the first email. This process yielded 15 topic-based datasets that contain between 135 and 1200 email dialogues. Owing to time limitations, the procedures described in this paper were applied to 8 of the datasets, corresponding to approximately 75% of the dialogues.

Analysis of our corpus yields the following observations.

- **O1:** Requests containing precise information, such as product names or part specifications, sometimes elicit helpful, precise answers referring to this information, while other times they elicit answers that do not refer to the query terms, but contain generic information (e.g., referring customers to another help group or asking them to call a particular phone number). Request-answer pair **RA1** in Figure 1 illustrates the first situation, while the pair **RA2** illustrates the second.[2]

[1] http://customercare.telephonyonline.com/ar/telecom_next_generation_customer.

[2] Our examples are reproduced verbatim from the corpus (except for URLs and phone numbers which have been disguised by us), and some have user or operator errors.

Proceedings of the Workshop on Task-Focused Summarization and Question Answering, pages 40–47,
Sydney, July 2006.

RA1:
Do I need Compaq driver software for my armada 1500 docking station? This in order to be able to re-install win 98?
I would recommend to install the latest system rompaq, on the laptop and the docking station. Just select the model of computer and the operating system you have. `http://www.thislink.com`.

RA2:
Is there a way to disable the NAT firewall on the Compaq CP-2W so I don't get a private ip address through the wireless network?
Unfortunately, you have reached the incorrect eResponse queue for your unit. Your device is supported at the following link, or at `888-phone-number`. We apologize for the inconvenience.

Figure 1: Sample request-answer pairs.

- **O2:** Operators tend to re-use the same sentences in different responses. This is partly a result of companies having in-house manuals that prescribe how to generate an answer. For instance, answers **A3** and **A4** in Figure 2 share the sentence in italics.

These observations prompt us to consider complementary approaches along two separate dimensions of our problem. The first dimension pertains to the *technique applied to determine the information in an answer*, and the second dimension pertains to the *granularity of the information*.

- Observation **O1** leads us to consider two techniques for obtaining information: *retrieval* and *prediction*. Retrieval returns an information item by matching its terms to query terms (Salton and McGill, 1983). Hence, it is likely to obtain precise information if available. In contrast, prediction uses features of requests and responses to select an information item. For example, the absence of a particular term in a request may be a good predictive feature (which cannot be considered in traditional retrieval). Thus, prediction could yield replies that do not match particular query terms.
- Observation **O2** leads us to consider two levels of granularity: *document* and *sentence*. That is, we can obtain a document comprising a complete answer on the basis of a request (i.e., reuse an answer to a previous request), or we can obtain individual sentences and then combine them to compose an answer, as is done in multi-document summarization (Filatova and Hatzivassiloglou, 2004). The sentence-level granularity enables the re-use of a sentence for different responses, as well as the composition of partial responses.

A3:
If you are able to see the Internet then it sounds like it is working, you may want to get in touch with your IT department to see if you need to make any changes to your settings to get it to work. *Try performing a soft reset, by pressing the stylus pen in the small hole on the bottom left hand side of the Ipaq and then release.*

A4:
I would recommend doing a soft reset by pressing the stylus pen in the small hole on the left hand side of the Ipaq and then release. Then charge the unit overnight to make sure it has been long enough and then see what happens. If the battery is not charging then the unit will need to be sent in for repair.

Figure 2: Sample answers that share a sentence.

The methods developed on the basis of these two dimensions are: *Retrieve Answer, Predict Answer, Predict Sentences, Retrieve Sentences* and *Hybrid Predict-Retrieve Sentences*. The first four methods represent the possible combinations of information-gathering technique and level of granularity; the fifth method is a hybrid where the two information-gathering techniques are applied at the sentence level. The generation of responses under these different methods combines different aspects of document retrieval, question-answering, and multi-document summarization.

Our aim in this paper is to investigate when the different methods are applicable, and whether individual methods are uniquely successful in certain situations. For this purpose, we decided to assign a level of success not only to complete responses, but also to partial ones (obtained with the sentence-based methods). The rationale for this is that we believe that a partial high-precision response is better than no response, and better than a complete response that contains incorrect information. We plan to test these assumptions in future user studies.

The rest of this paper is organized as follows. In the next section, we describe our five methods, followed by the evaluation of their results. In Section 4, we discuss related research, and then present our conclusions and plans for future work.

2 Information-gathering Methods

2.1 Retrieve a Complete Answer

This method retrieves a complete document (answer) on the basis of request lemmas. We use cosine similarity to determine a retrieval score, and use a minimal retrieval threshold that must be surpassed for a response to be accepted.

We have considered three approaches to indexing the answers in our corpus: according to the content lemmas in (1) requests, (2) answers, or (3) requests&answers. The results in Section 3 are for the third approach, which proved best. To illustrate the difference between these approaches, consider request-answer pair **RA2**. If we received a new request similar to that in **RA2**, the answer in **RA2** would be retrieved if we had indexed according to requests or requests&answers. However, if we had indexed only on answers, then the response would not be retrieved.

2.2 Predict a Complete Answer

This prediction method first groups similar answers in the corpus into answer clusters. For each request, we then predict an answer cluster on the basis of the request features, and select the answer that is most representative of the cluster (closest to the centroid). This method would predict a group of answers similar to the answer in **RA2** from the input lemmas "compaq" and "cp-2w".

The clustering is performed in advance of the prediction process by the intrinsic classification program *Snob* (Wallace and Boulton, 1968), using the content lemmas (unigrams) in the answers as features. The predictive model is a Decision Graph (Oliver, 1993) trained on (1) input features: unigram and bigram lemmas in the request,[3] and (2) target feature – the identifier of the answer cluster that contains the actual answer for the request.[4] The model provides a prediction of which response cluster is most suitable for a given request, as well as a level of confidence in this prediction. We do not attempt to produce an answer if the confidence is not sufficiently high.

In principle, rather than clustering the answers, the predictive model could have been trained on individual answers. However, on one hand, the dimensionality of this task is very high, and on the other hand, answers that share significant features would be predicted together, effectively acting as a cluster. By clustering answers in advance, we reduce the dimensionality of the problem, at the expense of some loss of information (since somewhat dissimilar answers may be grouped together).

2.3 Predict Sentences

This method looks at each answer sentence as though it were a separate document, and groups similar sentences into clusters in order to obtain meaningful sentence abstractions and avoid redundancy.[5] For instance, the last sentence in **A3** and the first sentence in **A4** are assigned to the same sentence cluster. As for Answer Prediction (Section 2.2), this clustering process also reduces the dimensionality of the problem.

Each request is used to predict promising clusters of answer sentences, and an answer is composed by extracting a sentence from such clusters. Because the sentences in each cluster originate in different response documents, the process of selecting them for a new response corresponds to multi-document summarization. In fact, our selection mechanism, described in more detail in (Marom and Zukerman, 2005), is based on a multi-document summarization formulation proposed by Filatova and Hatzivassiloglou (2004).

In order to be able to generate appropriate answers in this manner, the sentence clusters should be *cohesive*, and they should be predicted with high confidence. A cluster is cohesive if the sentences in it are similar to each other. This means that it is possible to obtain a sentence that represents the cluster adequately (which is not the case for an uncohesive cluster). A high-confidence prediction indicates that the sentence is relevant to many requests that share certain regularities. Owing to these requirements, the Sentence Prediction method will often produce partial answers (i.e., it will have a high precision, but often a low recall).

2.3.1 Sentence clustering

The clustering is performed by applying *Snob* using the following sentence-based and word-based features, all of which proved significant for

[3] Significant bigrams are obtained using the NSP package (`http://www.d.umn.edu/~tpederse/nsp.html`).

[4] At present, the clustering features differ from the prediction features because these parts of the system were developed at different times. In the near future, we will align these features.

[5] We did not cluster request sentences, as requests are often ungrammatical, which makes it hard to segment them into sentences, and the language used in requests is more diverse than the corporate language used in responses.

at least some datasets. The sentence-based features are:

- Number of syntactic phrases in the sentence (e.g., prepositional, subordinate) – gives an idea of sentence complexity.
- Grammatical mood of the main clause (5 states: imperative, imperative-step, declarative, declarative-step, unknown) – indicates the function of the sentence in the answer, e.g., an isolated instruction, part of a sequence of steps, part of a list of options.
- Grammatical person in the subject of the main clause (4 states: first, second, third, unknown) – indicates the agent (e.g., organization or client) or patient (e.g., product).

The word-based features are binary:

- Significant lemma bigrams in the subject of the main clause and in the "augmented" object in the main clause. This is the syntactic object if it exists or the subject of a prepositional phrase in an imperative sentence with no object, e.g., "click on *the following link*."
- The verbs in the sentence and their polarity (asserted or negated).
- All unigrams in the sentence, excluding verbs.

2.3.2 Calculation of cluster cohesion

To measure the textual cohesion of a cluster, we inspect the centroid values corresponding to the word features. Due to their binary representation, the centroid values correspond to probabilities of the words appearing in the cluster. Our measure is similar to entropy, in the sense that it yields non-zero values for extreme probabilities (Marom and Zukerman, 2005). It implements that idea that a cohesive group of sentences should agree strongly on both the words that appear in these sentences and the words that are omitted. Hence, it is possible to obtain a sentence that adequately represents a cohesive sentence cluster, while this is not the case for a loose sentence cluster. For example, the italicized sentences in **A3** and **A4** belong to a highly cohesive sentence cluster (0.93), while the opening answer sentence in **RA1** belongs to a less cohesive cluster (0.7) that contains diverse sentences about the Rompaq power management.

2.3.3 Sentence-cluster prediction

Unlike Answer Prediction, we use a Support Vector Machine (SVM) for predicting sentence clusters. A separate SVM is trained for each sentence cluster, with unigram and bigram lemmas in a request as input features, and a binary target feature specifying whether the cluster contains a sentence from the response to this request.

During the prediction stage, the SVMs predict zero or more clusters for each request. One representative sentence (closest to the centroid) is then extracted from each highly cohesive cluster predicted with high confidence. These sentences will appear in the answer (at present, these sentences are treated as a set, and are not organized into a coherent reply).

2.4 Retrieve Sentences

As for Sentence Prediction (Section 2.3), this method looks at each answer sentence as though it were a separate document. For each request sentence, we retrieve candidate answer sentences on the basis of the match between the content lemmas in the request sentence and the answer sentence. For example, while the first answer sentence in **RA1** might match the first request sentence in **RA1**, an answer sentence from a different response (about re-installing Win98) might match the second request sentence. The selection of individual text units from documents implements ideas from question-answering approaches.

We are mainly interested in answer sentences that "cover" request sentences, i.e., the terms in the request should appear in the answer. Hence, we use *recall* as the measure for the goodness of a match, where recall is defined as follows.

$$recall = \frac{\text{TF.IDF of lemmas in request sent \& answer sent}}{\text{TF.IDF of lemmas in request sentence}}$$

We initially retain the answer sentences whose recall exceeds a threshold.[6]

Once we have the set of candidate answer sentences, we attempt to remove redundant sentences. This requires the identification of sentences that are similar to each other — a task for which we use the sentence clusters described in Section 2.3. Again, this redundancy-removal step essentially casts the task as multi-document summarization. Given a group of answer sentences that belong to

[6]To assess the goodness of a sentence, we experimented with *f-scores* that had different weights for recall and precision. Our results were insensitive to these variations.

the same cohesive cluster, we retain the sentence with the highest recall (in our current trials, a cluster is sufficiently cohesive for this purpose if its cohesion ≥ 0.7). In addition, we retain all the answer sentences that do not belong to a cohesive cluster. All the retained sentences will appear in the answer.

2.5 Hybrid Predict-Retrieve Sentences

It is possible that the Sentence Prediction method predicts a sentence cluster that is not sufficiently cohesive for a confident selection of a representative sentence, but instead the ambiguity can be resolved through cues in the request. For example, selecting between a group of sentences concerning the installation of different drivers might be possible if the request mentions a specific driver. Thus the Sentence Prediction method is complemented with the Sentence Retrieval method to form a hybrid, as follows.

- For highly cohesive clusters predicted with high confidence, we select a representative sentence as before.
- For clusters with medium cohesion predicted with high confidence, we attempt to match the sentences with the request sentences, using the Sentence Retrieval method but with a lower recall threshold. This reduction takes place because the high prediction confidence provides a guarantee that the sentences in the cluster are suitable for the request, so there is no need for a convervative recall threshold. The role of retrieval is now to select the sentence whose content lemmas best match the request.
- For uncohesive clusters or clusters predicted with low confidence, we have to resort to word matches, which means reverting to the higher, more convervative recall threshold, because we no longer have the prediction confidence.

3 Evaluation

As mentioned in Section 1, our corpus was divided into topic-based datasets. We have observed that the different datasets lend themselves differently to the various information-gathering methods described in the previous section. In this section, we examine the overall performance of the five methods across the corpus, as well as their performance for different datasets.

3.1 Measures

We are interested in two performance indicators: *coverage* and *quality*.

Coverage is the proportion of requests for which a response can be generated. The various information gathering methods presented in the previous section have acceptance criteria that indicate that there is some level of confidence in generating a response. A request for which a planned response fails to meet these criteria is not covered, or addressed, by the system. We are interested in seeing if the different methods are applicable in different situations, that is, how exclusively they address different requests. Note that the sentence-based methods generate partial responses, which are considered acceptable so long as they contain at least one sentence generated with high confidence. In many cases these methods produce obvious and non-informative sentences such as "Thank you for contacting HP", which would be deemed an acceptable response. We have manually excluded such sentences from the calculation of coverage, in order to have a more informative comparison between the different methods.

Ideally, the **quality** of the generated responses should be measured through a user study, where people judge the correctness and appropriateness of answers generated by the different methods. However, we intend to refine our methods further before we conduct such a study. Hence, at present we rely on a text-based quantitative measure. Our experimental setup involves a standard 10-fold validation procedure, where we repeatedly train on 90% of a dataset and test on the remaining 10%. We then evaluate the quality of the answers generated for the requests in each test split, by comparing them with the actual responses given by the help-desk operator for these requests.

We are interested in two quality measures: (1) the precision of a generated response, and (2) its overall similarity to the actual response. The reason for this distinction is that the former does not penalize for a low recall — it simply measures how correct the generated text is. As stated in Section 1, a partial but correct response may be better than a complete response that contains incorrect units of information. On the other hand, more complete responses are favoured over partial ones, and so we use the second measure to get an overall indication of how correct and complete a response is. We use the traditional Information

Table 1: Performance of the different methods, measured as coverage, precision and f-score.

Method	Coverage	Precision Ave (stdev)	F-score Ave (stdev)
Answer Retrieval	43%	0.37 (0.34)	0.35 (0.33)
Answer Prediction	29%	0.82 (0.21)	0.82 (0.24)
Sentence Prediction	34%	0.94 (0.13)	0.78 (0.18)
Sentence Retrieval	9%	0.19 (0.19)	0.12 (0.11)
Sentence Hybrid	43%	0.81 (0.29)	0.66 (0.25)
Combined	72%	0.80 (0.25)	0.50 (0.33)

Retrieval precision and f-score measures (Salton and McGill, 1983), employed on a word-by-word basis, to evaluate the quality of the generated responses.[7]

3.2 Results

Table 1 shows the overall results obtained using the different methods. We see that combined the different methods can address 72% of the requests. That is, at least one of these methods can produce some non-empty response to 72% of the requests. Looking at the individual coverages of the different methods we see that they must be applicable in different situations, because the highest individual coverage is 43%.

The Answer Retrieval method addresses 43% of the requests, and in fact, about half of these (22%) are uniquely addressed by this method. However, in terms of the quality of the generated response, we see that the performance is very poor (both precision and f-score have very low averages). Nevertheless, there are some cases where this method uniquely addresses requests quite well. In three of the datasets, Answer Retrieval is the only method that produces good answers, successfully addressing 15-20 requests (about 5% of the requests in these datasets). These requests include several cases similar to **RA2**, where the request was sent to the wrong place. We would expect Answer Prediction to be able to handle such cases as well. However, when there are not enough similar cases in the dataset (as is the case with the three datasets referred to above), Answer Prediction is not able to generalize from them, and therefore we can only rely on a new request closely matching an old request or an old answer.

The Answer Prediction method can address 29% of the requests. Only about a tenth of these are uniquely addressed by this method, but the generated responses are of a fairly high quality, with an average precision and f-score of 0.82. Notice the large standard deviation of these averages, suggesting a somewhat inconsistent behaviour. This is due to the fact that this method gives good results only when complete template responses are found. In this case, any re-used response will have a high similarity to the actual response. However, when this is not the case, the performance degrades substantially, resulting in inconsistent behaviour. This behaviour is particularly prevalent for the "product replacement" dataset, which comprises 18% of the requests. The vast majority of the requests in this dataset ask for a return shipping label to be mailed to the customer, so that he or she can return a faulty product. Although these requests often contain detailed product descriptions, the responses rarely refer to the actual products, and often contain the following generic answer.

A5:
Your request for a return airbill has been received and has been sent for processing. Your replacement airbill will be sent to you via email within 24 hours.

Answer Retrieval fails in such cases, because each request has precise information about the actual product, so a new request can neither match an old request (about a different product) nor can it match the generic response. In contrast, Answer Prediction can ignore the precise information in the request, and infer from the mention of a shipping label that the generic response is appropriate. When we exclude this dataset from the calculations, both the average precision and f-score for the Answer Prediction method fall below those of the Sentence Prediction and Hybrid methods. This means that Answer Prediction is suitable when requests that share some regularity receive a complete template answer.

The Sentence Prediction method can find reg-

[7] We have also employed sequence-based measures using the ROUGE tool set (Lin and Hovy, 2003), with similar results to those obtained with the word-by-word measure.

ularities at the sub-document level, and therefore deal with cases when partial responses can be generated. It produces such responses for 34% of the requests, and does so with a consistently high precision (average 0.94, standard deviation 0.13). Only an overall 1% of the requests are uniquely addressed by this method, however, for the cases that are shared between this method and other ones, it is useful to compare the actual quality of the generated response. In 5% of the cases, the Sentence Prediction method either uniquely addresses requests, or jointly addresses requests together with other methods but has a higher f-score. This means that in some cases a partial response has a higher quality than a complete one.

Like the document-level Answer Retrieval method, the Sentence Retrieval method performs poorly. It is difficult to find an answer sentence that closely matches a request sentence, and even when this is possible, the selected sentences tend to be different to the ones used by the help-desk operators, hence the low precision and f-score. This is discussed further below in the context of the Sentence Hybrid method.

The Sentence Hybrid method extends the Sentence Prediction method by employing sentence retrieval as well, and thus has a higher coverage (45%). In fact, the retrieval component serves to disambiguate between groups of candidate sentences, thus enabling more sentences to be included in the generated response. This, however, is at the expense of precision, as we also saw for the pure Sentence Retrieval method. Although retrieval selects sentences that match closely a given request, this selection can differ from the "selections" made by the operator in the actual response. Precision (and hence f-score) penalizes such sentences, even when they are more appropriate than those in the model response. For example, consider request-answer pair **RA6**. The answer is quite generic, and is used almost identically for several other requests. The Hybrid method almost reproduces this answer, replacing the first sentence with **A7**. This sentence, which matches more request words than the first sentence in the model answer, was selected from a sentence cluster that is not highly cohesive, and contains sentences that describe different reasons for setting up a repair (the matching word in **A7** is "screen"). The Hybrid method outperforms the other methods in about 10% of the cases, where it either uniquely addresses requests, or addresses them jointly with other methods but produces responses with a higher f-score.

RA6:
My screen is coming up reversed (mirrored). There must be something loose electronically because if I put the stylus in it's hole and move it back and forth, I can get the screen to display properly momentarily. Please advise where to send for repairs.

To get the iPAQ serviced, you can call `1-800-phone-number`, options 3, 1 (enter a 10 digit phone number), 2. Enter your phone number twice and then wait for the routing center to put you through to a technician with Technical Support. They can get the unit picked up and brought to our service center.

A7:

To get the iPAQ repaired (battery, stylus lock and screen), please call `1-800-phone-number`, options 3, 1 (enter a 10 digit phone number), 2.

3.3 Summary

In summary, our results show that each of the different methods is applicable in different situations, all occurring significantly in the corpus, with the exception of the Sentence Retrieval method. The Answer Retrieval method uniquely addresses a large portion of the requests, but many of its attempts are spurious, thus lowering the combined overall quality shown at the bottom of Table 1 (average f-score 0.50), calculated by using the best performing method for each request. The Answer Prediction method is good at addressing situations that warrant complete template responses. However, its confidence criteria might need refining to lower the variability in quality. The combined contribution of the sentence-based methods is substantial (about 15%), suggesting that partial responses of high precision may be better than complete responses with a lower precision.

4 Related Research

There are very few reported attempts at corpus-based automation of help-desk responses. The retrieval system *eResponder* (Carmel et al., 2000) is similar to our Answer Retrieval method, where the system retrieves a list of request-response pairs and presents a ranked list of responses to the user. Our results show that due to the repetitions in the responses, multi-document summarization can be used to produce a single (possibly partial) representative response. This is recognized by Berger and Mittal (2000), who employ query-relevant summarization to generate responses. However, their corpus consists of FAQ

request-response pairs — a significantly different corpus to ours in that it lacks repetition and redundancy, and where the responses are not personalized. Lapalme and Kosseim (2003) propose a retrieval approach similar to our Answer Retrieval method, and a question-answering approach, but applied to a corpus of technical documents rather than request-response pairs. The methods presented in this paper combine different aspects of document retrieval, question-answering and multi-document summarization, applied to a corpus of repetitive request-response pairs.

5 Conclusion and Future Work

We have presented four basic methods and one hybrid method for addressing help-desk requests. The basic methods represent the four ways of combining level of granularity (sentence and document) with information-gathering technique (prediction and retrieval). The hybrid method applies prediction possibly followed by retrieval to information at the sentence level. The results show that with the exception of Sentence Retrieval, the different methods can address a significant portion of the requests. A future avenue of research is thus to characterize situations where different methods are applicable, in order to derive decision procedures that determine the best method automatically. We have also started to investigate an intermediate level of granularity: paragraphs.

Our results suggest that the automatic evaluation method requires further consideration. As seen in Section 3, our f-score penalizes the Sentence Prediction and Hybrid methods when they produce good answers that are more informative than the model answer. As mentioned previously, a user study would provide a more conclusive evaluation of the system, and could be used to determine preferences regarding partial responses.

Finally, we propose the following extensions to our current implementation. First, we would like to improve the representation used for clustering, prediction and retrieval by using features that incorporate word-based similarity metrics (Pedersen et al., 2004). Secondly, we intend to investigate a more focused sentence retrieval approach that utilizes syntactic matching of sentences. For example, if a sentence cluster is strongly predicted by a request, but the cluster is uncohesive because of a low verb agreement, then the retrieval should favour the sentences whose verbs match those in the request.

Acknowledgments

This research was supported in part by grant LP0347470 from the Australian Research Council and by an endowment from Hewlett-Packard. The authors also thank Hewlett-Packard for the extensive help-desk data, and Tony Tony for assistance with the sentence-segmentation software, and Kerri Morgan and Michael Niemann for developing the syntactic feature extraction code.

References

A. Berger and V.O. Mittal. 2000. Query-relevant summarization using FAQs. In *ACL2000 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 294–301, Hong Kong.

D. Carmel, M. Shtalhaim, and A. Soffer. 2000. eResponder: Electronic question responder. In *CoopIS '02: Proceedings of the 7th International Conference on Cooperative Information Systems*, pages 150–161, Eilat, Israel.

E. Filatova and V. Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *COLING'04 – Proceedings of the 20th International Conference on Computational Linguistics*, pages 397–403, Geneva, Switzerland.

G. Lapalme and L. Kosseim. 2003. Mercure: Towards an automatic e-mail follow-up system. *IEEE Computational Intelligence Bulletin*, 2(1):14–18.

C.Y. Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada.

Y. Marom and I. Zukerman. 2005. Towards a framework for collating help-desk responses from multiple documents. In *Proceedings of the IJCAI05 Workshop on Knowledge and Reasoning for Answering Questions*, pages 32–39, Edinburgh, Scotland.

J.J. Oliver. 1993. Decision graphs – an extension of decision trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pages 343–350, Fort Lauderdale, Florida.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity – measuring the relatedness of concepts. In *AAAI-04 – Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 25–29, San Jose, California.

G. Salton and M.J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw Hill.

C.S. Wallace and D.M. Boulton. 1968. An information measure for classification. *The Computer Journal*, 11(2):185–194.