Automatic handling of Frequently Asked Questions using Latent Semantic Analysis

Patrik Larsson and **Arne Jönsson**
Department of Computer and Information Science
Linköping University, SE-581 83, LINKÖPING, SWEDEN
patla073@gmail.com, arnjo@ida.liu.se

Abstract

We present results from using Latent Semantic Analysis (LSA) for automatic handling of FAQs (Frequently Asked Questions). FAQs have a high language variability and include a mixture of technical and non-technical terms. LSA has a potential to be useful for automatic handling of FAQ as it reduces the linguistic variability and capture semantically related concept. It is also easy to adapt for FAQ. LSA does not require any sophisticated linguistic analyses and merely involves various vector operations. We evaluate LSA for FAQ on a corpus comprising 4905 FAQ items from a collection of 65000 mail conversations. Our results show that Latent Semantic Analysis, without linguistic analyses, gives results that are on par other methods for automatic FAQ.

Introduction

Automatic FAQ-systems allow clients' requests for guidance, help or contact information to be handled without human intervention, c.f. (Åberg 2002). Typically, automatic FAQ-systems use previously recorded FAQ-items and various techniques to identify the FAQ-item(s) that best resembles the current question and present a matching answer. For instance, the FAQFinder system (Mlynarczyk and Lytinen 2005) uses existing FAQ knowledge bases to retrieve answers to natural language questions. FAQFinder utilises a mixture of semantic and statistical methods for determining question similarities. Another technique is to use a frequency-based analysis from an ordinary FAQ list with given/static questions and answers (Ng'Ambi 2002). Linguistic based automatic FAQ systems often starts with finding the question word, keywords, keyword heuristics, named entity recognition, and so forth (Moldovan et al. 1999). Another approach is to use machine learning techniques, such as support vector machines to predict an appropriate response (Marom and Zukerman 2007; Bickel and Scheffer 2004). Marom and Zukerman also utilise a variety of clustering techniques to produce more accurate answers.

One issue for automatic help-desk systems is that we often have many-to-many mappings between requests and responses. A question is stated in many ways and, as humans answer the requests, the response to a question can be stated in many ways. The propositional content can also vary, although operators re-use sentences, at least in e-mail help desk-systems (Zukerman and Marom 2006).

Help-desk e-mail conversations are further characterised by: (1) having many requests raising multiple issues, (2) having high language variability and (3) with many answers utilising non-technical terms not matching technical terms in the requests (Marom and Zukerman 2007).

In this paper we present results from experiments on using linear algebra techniques for automatic FAQ for single issues. We will not consider (1), i.e. we will not present answers to requests comprising multiple issues. Our study is based on a log of email dialogues between customers and help-desk operators at Hewlett-Packard (Marom and Zukerman 2007)[1]. A typical example is seen in Figure 1. The dialogues deal with a variety of issues such as technical assistance, Figure 1, product information, and product replacement.

Latent Semantic Analysis

To handle the FAQ problem of having high language variability and a mixture of non-technical terms and technical terms, it is natural to investigate techniques that in various ways reduce the linguistic variability and better capture semantically related concepts. One prominent such technique is vector space models, for an overview see Eldén (2007). The basic idea is to formulate the problem in a matrix terminology, usually by constructing a matrix using text units as columns and letting the words in all text units each have a row in the matrix. Then a certain entry in the matrix is nonzero iff the word corresponding to the row exists in the text unit represented by the column. The resulting matrix is very large and sparse. Techniques for reducing dimensionality are therefore often used.

Latent Semantic Analysis (LSA) (Landauer and Dumais 1997) is one of the most well-known dimensionality reduction vector space techniques and has also been used on a limited help-desk data set with promising results (Caron 2000).

LSA uses Singular Value Decomposition (SVD) to find a reduced vector space that fits the original as well as possible using a lower ranked matrix. The SVD factorizes a matrix A

[1]We are indebted to HP, Yuval Marom and Ingrid Zukerman for providing this corpus.

User question:
When I start my computer from time to time I get the following error message
""Terminating thread due to stacking overflow problem. A VxD, possibly recently installed, has consumed too much space.
Increase setting of MinSPs' in System.INI or remove recent VxD's.
4 SP's installed""
Please tell me how to make the change in number SP's installed to correct this problem.
Thanks, FirstName
Model answer:
RE: Armada 1500 P-prtbl-dg" "Dear HP Customer,
Thank you for contacting HP eServices.
This is ni response to your e-mail regarding the error message on your Armada 1500 Notebook.
Please be informed that Stack overflow conditions are indicative of errors in the device driver. Device drivers are allocated 4 kilobytes (one page) of space to be used as a stack. If the device driver uses more than 4 kilobytes of memory, a stack overflow condition occurs.
In order to resolve the issue we recommend following the steps given below.
A long list of instructions follows, which are removed.
We recommend visiting the following link for more information on the error message.
Link removed
Please e-mail us, if you need any further assistance and we will be glad to help.
Thank you,
HP eServices

Figure 1: Example of question and answer from the HP corpus.

into $A = USV^T$ where V^T is the new orthonormal basis for A, S is a diagonal matrix denoting how prevalent each column is in the basis while U is the coordinates of the original documents using the new basis.

Documents can then be compared in the new vector space, by changing the basis of the document to V^T and then compare likeness with the coordinates in U, often based on the cosine between the two.

LSA resolves problems with synonymy, polysemy, homonymy etc. by mapping (or mixing) terms occuring often in the same context to each other (Landauer et al. 2007).

For automatic FAQ systems LSA directly allows for mappings between combinations of questions and answers. The first-order relations in this domain are:

- Terms in questions – Terms in similar questions
- Terms in questions – Terms in questions with similar responses
- Terms in responses – Terms in similar responses
- Terms in responses – Terms in responses with similar questions

A "request term" like *power coord* and similar terms used in other requests will be mapped to the technical term *AC-adapter* used in responses by the helpdesk-support personnel. "Request terms" like *strange, blinking, green, light* will be mapped to the terms in other requests or responses resolving the issue at hand.

LSA also captures higher-order relations between terms, and thus create mappings between terms that do not directly co-occur, but that mutually co-occur with other terms.

LSA for automatic FAQ

When performing LSA on FAQs, a Question-Answer item (QA-item), such as Figure 1 in the corpus corresponds to a document. In the corpus questions and answers are simple text files with indicators for question and answer. The corpus comprise two-turn dialogues as well as longer dialogues with follow-up questions. Just as Marom and Zukerman 2007 we only used two-turn dialogues with reasonably concise answers (16 lines at most).

The vector space is constructed by having all QA-items in the matrix on one axis and all the words on the other and then calculate the frequency of the words in relation to the QA-items. Questions and answers are not separated in the QA-items. In our case we create the $m \times n$ matrix A where each matrix element a_{ij} is the weight of word i in document j. The n columns of A represent the QA-items in the corpus and the rows correspond to the words, as seen in Figure 2. We use 4414 QA-items for training comprising 35600 words. The size of the original training matrix A is thus 4414×35600. About 524900 elements of the total 157 millions are nonzero[2].

	QA_1	QA_2	...	QA_n
word_1	a_{11}	a_{12}	...	a_{1n}
word_2	a_{21}	a_{22}	...	a_{2n}
...	...	...	...	...
word_m	a_{m1}	a_{m2}	...	a_{mn}

Figure 2: A word–QA-item matrix

Performing SVD on A with dimension k produces three

[2]The exact numbers depend on which subset of QA-items that are used for training.

new matrices, the $m \times k$ matrix U which corresponds to the QA-items in the reduced vector space, the $k \times k$ diagonal matrix S and the $k \times n$ matrix V which corresponds to the words in the reduced vector space. The size of the matrix U depends on the dimension, k. For a reduced vector space with $k = 300$ it is 35600×300 and the size of the matrix V^T is 300×4414.

In order to do LSA for automatic FAQ we perform the following steps:

1. Pre-process all Question-Answer items in the corpus, Section "Pre-processing".
2. Perform Singular Value Decomposition on the matrix A_{tr} obtained from the set of QA-items in the training set. This gives us the three components U_{tr}, S_{tr} and V_{tr}^T.
3. Fold in the answers from the training set of QA-items into the set of vectors in U_{tr}, Section "Folding Questions and Answers into LSA space", Equation 4. This gives us a new matrix, U_{folded}, i.e. a *pseudo-document* with all answers folded into the reduced vector space.
4. Create answer clusters, $A_{cluster}$, from U_{folded} using QT-clustering, Section "Answer clustering".
5. Create a new matrix of tagged left singular vectors U_{tagged} by using the clusters $A_{cluster}$ to tag U_{tr} and remove items that do not belong to any cluster. Select a representative answer from each cluster, Section "Selecting a representative answer from a faq-cluster".
6. Fold in questions one by one from the test set, Section "Folding Questions and Answers into LSA space", Equation 5. Compare to the tagged matrix of left singular vectors U_{tagged}, see Section "Answer clustering" and pick the best.

In what follows we will describe each step in more detail.

Pre-processing

The QA-items are used without any linguistic pre-processing, i.e. we use no stop-word lists, stemming, etc (Landauer et al. 2007). Nor any Named-Entity recognition or abbreviation lists.

The StandardAnalyzer in Lucene[3] is used for tokenization and vectorization, i.e. creating vectors from the tokens.

To reduce the impact of terms which are evenly distributed in the corpus, Question-Answer vectors are entropy normalised by using the global term weights from the matrix used for SVD, where a document consists of QA-items. These term weights are then used to weight the terms in the question and answer documents as follows (Gorrell 2006) :

$$p_{ij} = \frac{tf_{ij}}{gf_i} \tag{1}$$

$$gw_i = 1 + \sum_j \frac{p_{ij} log(p_{ij})}{log(n)} \tag{2}$$

$$c_{ij} = gw_i log(tf_{ij} + 1) \tag{3}$$

where c_{ij} is the cell at column i, row j in the corpus matrix and gw_i is the global weighting of the word at i, n is the number of QA-items. tf_j is the term frequency in document j and gf the global count of term i across all documents. Following Gorrell (2006) we use tf_{ij} instead of p_{ij} in Equation 3.

Performing Singular Value Decomposition

We use SVDLIBC[4] for singular value decomposition. Clustering and testing are performed in MatLab. SVDLIBC is used to generate the three components U_{tr}, S_{tr} and V_{tr}^T. We will investigate the influence of different dimensions, i.e. different reduced vector spaces.

We perform singular value decomposition on the training set of questions and answers. Answers are folded into the new, dimension reduced, vector space afterwards using Matlab, see Section "Folding Questions and Answers into LSA space".

The SVD components U_{tr}, S_{tr} and V_{tr}^T are imported to MatLab in Matlab ascii-format together with the files containing the training questions + training answers, the training answers, and later the test questions (and unique id-numbers for the dialogues).

Folding Questions and Answers into LSA space

Given that we have a vector space matrix A with QA-items and words but want to have answers as responses to requests we need to transform the answers into the reduced vector space. This is done by folding-in the answers into the reduced vector space model and produce a *pseudo-document* (Wang and Jin 2006).

The answers in the training corpus are folded into the reduced space after we performed SVD. This is in line with findings of Zukerman and Marom (Zukerman and Marom 2006) who find that using both questions and answers to retrieve an answer proved better than using only questions or answers.

Answers are folded in by taking the dot product of the vector with the reduced space right singular matrix, V_{tr}, i.e. the terms in the reduced vector space, Equation 4 (Gorrell 2006, p. 34).

$$\mathbf{a}_{folded} = \mathbf{a} \cdot V_{tr} \tag{4}$$

Taking all $\mathbf{a}_{folded}$ vectors creates U_{folded}, a *pseudo-document* representation of size $n(documents) \times k$ where k is the new reduced vector space dimension and $n(documents)$ is the number of terms in the QA-items, i.e. all unique words occuring in the questions and answers.

Similarly, questions in the test set need to be folded into the dimension reduced set of clustered answers, Equation 5, see Section "Classifying new questions".

$$\mathbf{q}_{folded} = \mathbf{q} \cdot V_{tr} \tag{5}$$

Folding in questions allows us to map questions to the reduced vector space.

[3] http://lucene.apache.org/

[4] http://tedlab.mit.edu/~dr/svdlibc/

Create a sparse matrix, M, with

$$m_{ij} = \begin{cases} 1 & \text{if } a_{ij} > \tau \\ 0 & \text{otherwise} \end{cases}$$

where
$a_{ij} = a_i \cdot a_j$
a is an answer in the *pseudo-document* with the folded-in answers, U_{folded}, $i \neq j$
and τ is the maximum cluster diameter

Extract clusters from M:

while max(row-sum)$> \gamma$
For each row i in M calculate:
$r_i = \sum_j a_{ij}$
Save the row with highest row sum as cluster c_k
Remove all rows and columns from M belonging to cluster c_k

Figure 3: QT-clustering algorithm

Answer clustering

One problem with automatic handling of FAQ is that in the corpus one question can have many correct answers, depending on the person answering the question. Each operator uses different wordings. They also provide varying amounts of information (Zukerman and Marom 2006). Consequently, we want to cluster answers that provide the same, or similar, information. In the future, the use of FAQ-databases would alleviate this problem somewhat.

One way to handle this is to cluster around QA-items. However, the "request domain" is a more open domain than the "answer domain", the "request domain" often contains irony and completely irrelevant information, for example:

```
   If this gives you a good laugh that's
OK but I'm serious and very desperate -
at least I know that the CD isn't a cup
holder...  The number I have given you
is my home phone number.  There's no way
I'll take this call at work.
```

The answers on the other hand contain very specific instructions, are more formal in style and are devoid of irony, c.f. Figure 1. Thus, we only cluster the dialogues based on the answers.

We use Quality Threshold Clustering (Heyer, Kruglyak, and Yooseph 1999) for answer clustering, see Figure 3, and use cosine-distances between normalized answer vectors as the maximum cluster diameter, τ.

γ controls the number of elements in each cluster. A low γ may result in small clusters where the similarity of the answer is accidental, for example a user who by mistake submits the same question twice may receive two identical replies, the cluster consisting of these replies would not represent responses to a frequently asked question but merely the fact that the question was sent twice. A too low limit on cluster size therefore increases the risk of not including a relevant answer.

Creating the adjacency matrix, M, can be somewhat computationally demanding, but as it can be done incrementally it poses no computational problems.

This clustering method guarantees that the LSA-similarity of frequent answer clusters will not exceed a predefined threshold, and this threshold is meaningful because LSA-similarity between documents have shown a high correlation with human judgment (Landauer, Laham, and Foltz 1998).

A similarity threshold, τ, of 0.6 - 0.9 is usually considered acceptable, but it depends on the specific domain. We will investigate the best threshold, τ, for the FAQ domain. A technical domain like helpdesk-support might need a larger threshold than more "soft domains", as the answers are less varied. A large threshold generates large clusters which has an advantage in that there will be more questions and therefore more mappings between questions and answers. There will be more members in each cluster and also more clusters as there are more members (i.e. answers) that can be nearest neighbour when classified. Thus, we achieve a higher Coverage. On the other hand, a too large threshold probably means a decrease in Precision and Recall.

Classifying new questions

To find the best answer cluster for a new request we use a basic k-nearest neighbour classifier (Cardoso-Cachopo and Oliveira 2003). We perform the following steps:

1. Find the distance for the new request to the dimension reduced QA-items by computing the dot product of the new request, $\mathbf{q}$, with all U_{tr} vectors, $\mathbf{u}$, in all clusters, i.e. all QA-items.
   $$a_i = \mathbf{q} \cdot \mathbf{u}_i$$
2. Pick the k nearest a_i, i.e. answers close to the QA-items´, $\mathbf{u}$.
3. Select the cluster with most a_i items and a representative from that cluster answer as above, Section "Selecting a representative answer from a faq-cluster".

kNN is used to exclude outliers, QA-items that accidentally are close to the new request, e.g. QA-items containing misspelled words that are misspelled the same way in the new request.

Using a more sophisticated classifier might improve performance somewhat, but a basic classifier like kNN generally gives good performance when combined with Latent Semantic Analysis (Cardoso-Cachopo and Oliveira 2003).

Selecting a representative answer from a faq-cluster

To select an answer document from the matched cluster we first normalize the answer vectors to minimize the influence of "flooded answers", that is, answers that contain relevant information, but a large portion of irrelevant information as well (for example an answer message containing responses to more than one question). We use standard length normalisation:

$$\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|} \tag{6}$$

A representative answer is then selected from, U_{tagged} using the cosine angel. Answers must exceed a threshold δ to be selected. This is done to ensure that they are not too far away from the request. We investigate two methods for selecting a representative answer. One method takes the answer closest to the centroid of the cluster as being the most representative answer. The other method takes the closest answer in the cluster. The former provides a more neutral answer and is probably not wrong but may not contain enough information. The latter, on the other hand, provides answers that may be wrong, but if correct probably convey more relevant information.

Evaluation

We have evaluated LSA for automatic FAQ on the corpus from HP with email messages between users and helpdesk operators. We use the same sub-corpus of 4,905 two-turn dialogues divided into 8 subsets as Marom and Zukerman (2007). In the experiments all 8 data sets are grouped into one large set. Typical for this test set is that answers are short (less than 16 lines). This was done to ensure that answers do not contain multiple answers etc. (Marom and Zukerman 2007). We use 4414 dialogues for training and 491 for testing.

We have conducted experiments to find optimal values of τ, SVD dimension and how to select an answer from the answer clusters; using the centroid in an answer cluster vs. taking the closest answer. We also study δ, k and the minimum cluster size γ.

We use the ROUGE tool set version 1.5.5 to produce Precision, Recall and F-scores for one-gram-overlaps (ROUGE-1). We apply equal weight to Recall and Precision when calculating F-scores. ROUGE then produces similar results as word-by-word measures (Marom and Zukerman 2007).

The term "Coverage" is used to measure the amount of a test set where any reply was given based on the threshold settings of the method used (Marom and Zukerman 2007).

Results and discussion

The output from the automatic FAQ system varies depending on from which data set an answer is retrieved. Some requests are answered using a short and fairly standardised answer which are easy to retrieve by the system. For instance finding answers to requests in the Product Replacement data set is mostly trivial, the documents are highly similar and can be matched on the basis of the title, Figure 5.

Other requests fail to produce an answer, or produce an empty response indicating that there are no answers close enough in the answer cluster. Many requests also produce correct, but not equal, answers as in Figure 6. In this case the answer contains more information than the original answer to the request did.

Parameter setting investigations

We have investigated the effect on different SVD dimensions, see Figure 7. As can be seen in Figure 7 the ROUGE-1 Precision, Recall and F-scores reach a maximum after a dimension of around 250 and stays the same up to around 650.

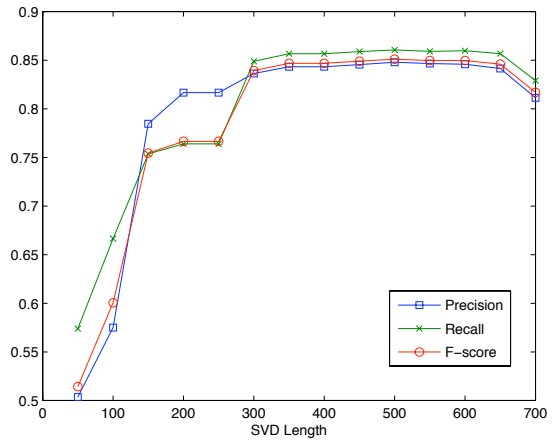


Figure 7: The influence of different SVD dimension reductions

We use the 300 first singular vectors as the gain from utilising a larger vector space does not motivate the longer processing time and increased memory usage needed for longer vectors. This is in line with previous findings by e.g. (Landauer and Dumais 1997).

The value on the threshold δ affects Coverage, normally we use $\delta = 0.6$ as this gives reasonable Coverage. When τ increases Coverage also increases. In the final investigation we use $\delta = 0.43$ when $\tau = 0.9$ to have the same Coverage of 29% as Marom and Zukerman (2007), see below.

We conducted experiments on the effect of using the centroid answer versus the answer closest to the request, max cosine. We did that for two different values of τ, as τ affect the number of answers in a cluster and consequently the centroid answer. Figure 4 shows values for Coverage, Precision, Recall and F-score for $\tau = 0.6$ and $\tau = 0.9$. Using the centroid answer gives Precision, Recall and F-scores that are higher than the corresponding values for closest answer for both values of τ. Coverage is slightly better, but that improvement does not justify the higher loss in Precision and Recall. We will, thus, use the centroid in the experiments presented below.

We have investigated the effect different values on k and γ have on Coverage, Precision and F-score, see Figure 9.

The parameters k in kNN and γ in QT-clustering covaries. To study them one by one we used a fix value, 1, for k when varying γ and $\gamma = 1$ when varying k. To reduce the risk of equal votes, we only use odd values for k.

As can be seen in Figure 9 increasing γ and k have some effect up until $\gamma = 7$ and $k = 5$, for $k = 1$ and $\gamma = 1$ respectively. We use $k = 5$ and $\gamma = 5$ in our experiments. The parameters co-vary and the exact values are not critical, as long as they are not too small.

The QT-clustering diameter, τ, is varied between 0.6 and 0.9 (Landauer, Laham, and Foltz 1998). For small τ we get a higher Coverage, and slightly lower Precision, but Precision is not that much affected for τ between 0.6 and 0.9, Figure 8.

To be more precise. The ROUGE-1 values for τ = 0.7,

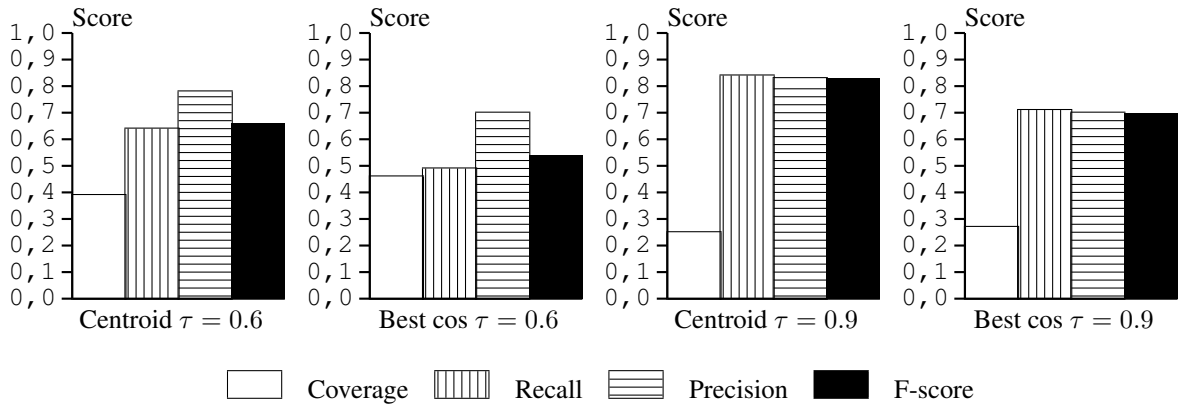


Figure 4: Answer selection based on the centroid vs the best cosine match.

Request, Q, to be answered	**Request corresponding to the answer A**
I need a replacement return shipping label P-eurp Need return shipping label and box for 30 gig laptop hard drive, p/n *Product number*	**I need a replacement return shipping label P-eurp** I had 4 separate power supplies sent to me and no longer have the boxes with the return labels. I need 4 return labels for the following case numbers including the one above. Thank you. *List of ID numbers*
Answer in FAQ base corresponding to Q	**Answer, A, generated to Q**
RE: I need a replacement return shipping label P-eurp Good Morning, I apologize for the delay in responding to your issue. 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. Thank You, E Services	**RE: I need a replacement return shipping label P-eurp** Good Morning, I apologize for the delay in responding to your issue. 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. Thank You, E Services

Figure 5: Example of a trivial response. The upper left side shows the new request to the automatic FAQ system, the lower left side shows the correct corresponding answer to that request. The lower right hand side shows the answer presented as a response to the new request, the request in the upper left, and the upper right shows the request that corresponds to the presented answer.

providing 35% Coverage, are:

Average_R: 0.71792 (95%-conf.int. 0.67195 - 0.76372)
Average_P: 0.79824 (95%-conf.int. 0.76347 - 0.83251)
Average_F: 0.72578 (95%-conf.int. 0.68454 - 0.76708)

Allowing a slightly higher Coverage of 40%, τ = 0.6, we still achieve acceptable results:

Average_R: 0.64948 (95%-conf.int. 0.60443 - 0.69585)
Average_P: 0.78956 (95%-conf.int. 0.75779 - 0.82156)
Average_F: 0.66915 (95%-conf.int. 0.62896 - 0.71028)

Comparisons to other approaches

Our results are better than the results obtained when using SVM only (Bickel and Scheffer 2004). For instance, for a Recall around 30% they never produce Precision above 0.7. However, they use a different corpus, so the results are not totally comparable.

It is not straightforward to compare our results with the results from Marom and Zukerman (2007). Marom and Zukerman present three different methods: *Doc-Pred, Sent-Pred* and *Sent-Hybrid*, and also a meta-learning method utilising combinations of the methods (Marom, Zukerman, and Japkowicz 2007). Basically *Doc-Pred* uses request features from the questions to retrieve the best complete answer (possibly comprising more than one sentence). *Sent-Pred* use the same technique to analyse the new request but here the answer is composed of sentences from different answers. *Sent-Hybrid*, finally, utilise terms in the requests and answers, to further improve performance.

We do not split answers into sentences, and consequently, we should not be able to reproduce the results from Marom and Zukerman (2007) where they produce answers com-

Request, Q, to be answered	Request corresponding to the answer A
iPAQ H3700 Series Pocket PC P-hand I am unable to download activesync 3.6. The Microsoft website is not working when I click the download button.	**iPAQ H3700 Series Pocket PC P-hand** I have a H3635 and the software that was on my computer to sync information from the computer to the cradle is now erased from my computer, how do I install new software on my computer? Can I install it from the internet site or do I have to purchase the CD from the store because I do not have the orginal one that came with my Pcket PC...Help me
Answer in FAQ base corresponding to Q	**Answer, A, generated to Q**
RE: iPAQ H3700 Series Pocket PC P-hand Dear Customer, Thank you for contacting HP eServices. Please visit the link given below and download Activesync 3.6: *web site* If you need any more clarifications, please email us back and we would be glad to help you. Regards, HP eServices	**RE: iPAQ H3700 Series Pocket PC P-hand** Dear HP Customer, Thank you for contacting HP eServices. If you need a synchronization software in order to synchronize the iPAQ with your Desktop PC, you could download the latest ActiveSync 3.6 from Microsoft"s website at *web site* For complete installation instructions of the above ActiveSync 3.6 software, please visit the link *web site* If you need any further clarifications, please e-mail us back and we would be glad to help. Thank you, HP eServices

Figure 6: Example of a correct but not identical response. The upper left side shows the new request to the automatic FAQ system, the lower left side shows the correct corresponding answer to that request. The lower right hand side shows the answer presented as a response to the new request, the request in the upper left, and the upper right shows the request that corresponds to the presented answer.

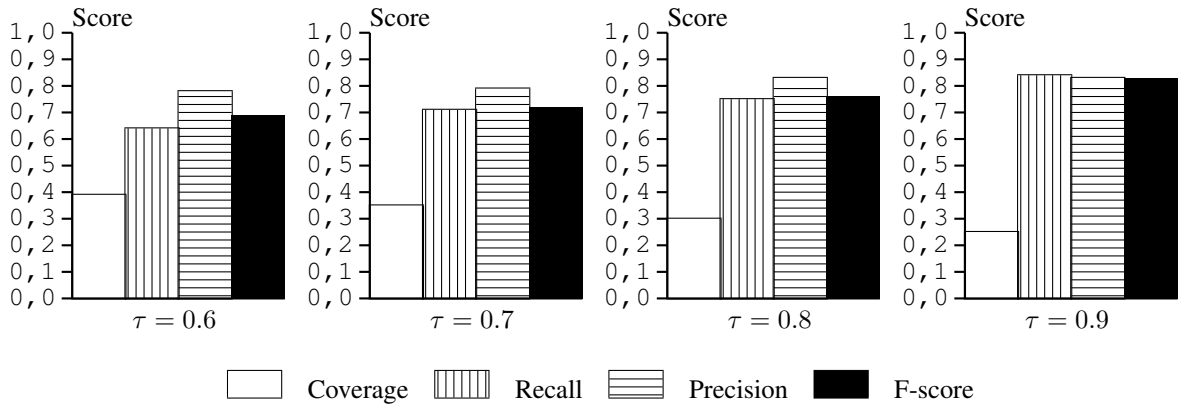


Figure 8: Comparison of different cluster diameters for $k = 5$ and $\delta = 0.6$.

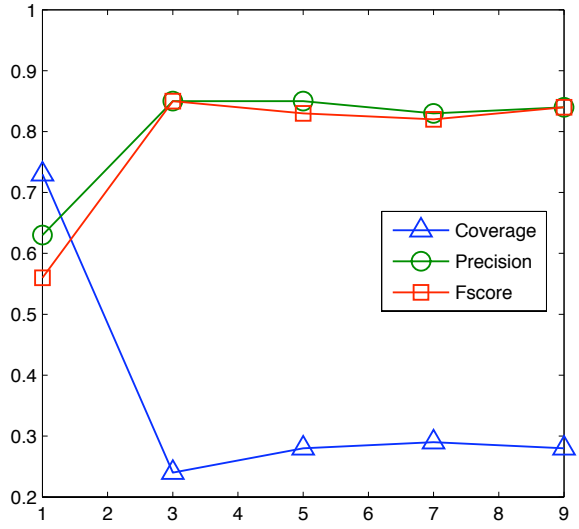
posed by sentences from different answers, *Sent-Pred, Sent-Hybrid*. Comparing our results with *Doc-Pred* we see that our results are similar to their results, Table 1[5].

To be more precise, we use τ = 0.9 and $\delta = 0.43$ to achieve 29% Coverage. With k = 5 we get the following ROUGE-1 values:

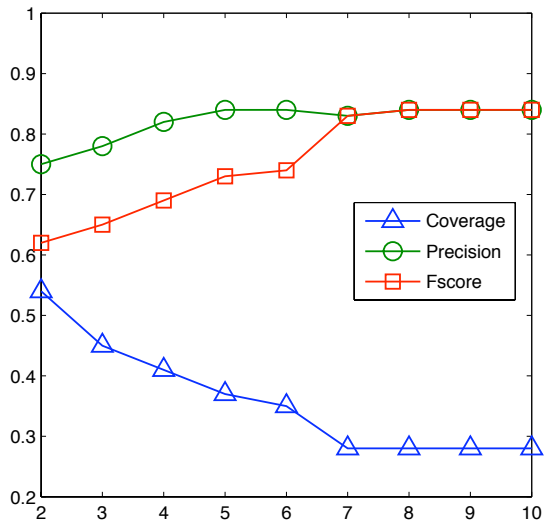
Average_R: 0.83814	(95%-conf.int. 0.80448 - 0.87163)
Average_P: 0.82713	(95%-conf.int. 0.79470 - 0.85939)
Average_F: 0.82726	(95%-conf.int. 0.79412 - 0.86162)

To further verify the results, we conducted a ten-fold evaluation on the whole corpus. This gave the following

[5]Values on *Doc-Pred* from Marom and Zukerman (2007).



k with $\gamma = 1$



γ with $k = 1$

Figure 9: The influence of different γ and k

	Coverage	Recall	F-score
Doc-Pred	29%	0.82	0.82
LSA	29%	0.83	0.83

Table 1: LSA compared to *Doc-Pred*

ROUGE-1 values (τ = 0.9, k = 5, 29% Coverage):

Average_R: 0.81231 (95%-conf.int. 0.79795 - 0.82571)
Average_P: 0.85251 (95%-conf.int. 0.84344 - 0.86194)
Average_F: 0.80643 (95%-conf.int. 0.79401 - 0.81870)

We see that the there is a small decrease in Recall and a slight increase in Precision. The reason for this is that there are a number of empty messages that give 100% Precision and 0% Recall. The results are, however, still on par with *Doc-Pred*.

Summary

In this paper we have presented results from using Latent Semantic Analysis for automatic FAQ handling. Using LSA is straightforward and requires very little domain knowledge or extra processing steps such as identifying terms, removing stop words, etc. All we do are standard vector operations, mainly in LSA space. Consequently, the method is easy to utilise in new domains.

Our results show that LSA is a promising method for automatic FAQ. The results are on a par with the *Doc-Pred* method of Marom and Zukerman (2007).

One problem with LSA is the computational demands of SVD. For practical applications it is possible to handle the computational problem with SVD by collecting Question-Answer pairs continuously and fold them into LSA space (clustering can be done incrementally), and update the SVD regularly (perhaps once a month) with "representative" Question-Answer pairs used for mapping new questions to the domain.

Another possibility is to perform the SVD incrementally by using Generalised Hebbian Learning (GHA) for SVD (Gorrell 2006). This allows for incremental SVD and handles very large data sets. Yet another possibility is to reduce the dimensionality of the matrix on which SVD is calculated using Random Indexing (Gorrell 2006; Kanerva, Kristofersson, and Holst 2000; Sellberg and Jönsson 2008).

Further work includes splitting up answers into sentences and perform answer clustering like *Sent-Hybrid* (Marom and Zukerman 2007). By using sentences instead of answers in our matrix we can form answer clusters.

We did not perform any pre-processing as suggested by Landauer et al. (2007). Named-Entity recognition can probably further improve the results in a final system.

Acknowledgment

This research is financed by Santa Anna IT Research Institute AB.

References

Bickel, S., and Scheffer, T. 2004. Learning from message pairs for automatic email answering. In Boulicaut, J.-F.; Esposito, F.; Giannotti, F.; and Pedreschi, D., eds., *Proceedings of Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004*, volume 3201 of *Lecture Notes in Computer Science*, 87–98. Springer.

Cardoso-Cachopo, A., and Oliveira, A. L. 2003. An empirical comparison of text categorization methods. In *Inter-*

national Symposium on String Processing and Information Retrieval, SPIRE, LNCS, volume 10.

Caron, J. 2000. Applying lsa to online customer support: A trial study. Master's thesis, University of Colorado, Boulder.

Eldén, L. 2007. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial & Applied Mathematics (SIAM).

Gorrell, G. 2006. *Generalized Hebbian Algorithm for Dimensionality Reduction in Natural Language Processing*. Ph.D. Dissertation, Linköping University.

Heyer, L. J.; Kruglyak, S.; and Yooseph, S. 1999. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research* 9(11):1106–1115.

Kanerva, P.; Kristofersson, J.; and Holst, A. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society. Erlbaum, 2000.*, 1036.

Landauer, T. K., and Dumais, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104:211–240.

Landauer, T. K.; McNamara, D.; Dennis, S.; and W., K., eds. 2007. *Handbook of Latent Semantic Analysis*. Mahwah NJ: Lawrence Erlbaum Associates.

Landauer, T. K.; Laham, D.; and Foltz, P. 1998. Learning human-like knowledge by singular value decomposition: A progress report. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Marom, Y., and Zukerman, I. 2007. A predictive approach to help-desk response generation. In Veloso, M. M., ed., *Proceedings of IJCAI 2007, Hyderabad, India*, 1665–1670.

Marom, Y.; Zukerman, I.; and Japkowicz, N. 2007. A meta-learning approach for selecting between response automation strategies in a help-desk domain. In *AAAI*, 907–912. AAAI Press.

Mlynarczyk, S., and Lytinen, S. 2005. Faqfinder question answering improvements using question/answer matching. In *Proceedings of L&T-2005 - Human Language Technologies as a Challenge for Computer Science and Linguistics*.

Moldovan, D. I.; Harabagiu, S. M.; Pasca, M.; Mihalcea, R.; Goodrum, R.; Girju, R.; and Rus, V. 1999. Lasso: A tool for surfing the answer net. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8), 1999.*

Ng'Ambi, D. 2002. Pre-empting user questions through anticipation: data mining faq lists. In *Proceedings of the 2002 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on Enablement Through Technology*. ACM International Conference Proceeding Series.

Sellberg, L., and Jönsson, A. 2008. Using random indexing to improve singular value decomposition for latent semantic analysis. In *Proceedings of the 6th Conference on Language Resources and Evaluation. Marrakech, Morocco*.

Wang, X., and Jin, X. 2006. Understanding and enhancing the folding-in method in latent semantic indexing. In Bressan, S.; Küng, J.; and Wagner, R., eds., *Database and Expert Systems Applications, 17th International Conference, DEXA 2006, Kraków, Poland, September 4-8, 2006, Proceedings*, volume 4080 of *Lecture Notes in Computer Science*, 104–113. Springer.

Zukerman, I., and Marom, Y. 2006. A comparative study of information-gathering approaches for answering help-desk email inquiries. In *Proceedings of 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia*.

Åberg, J. 2002. *Live Help Systems: An Approach to Intelligent Help for Web Information Systems*. Ph.D. Dissertation, Linköpings universitet, Thesis No 745. `http://www.ida.liu.se/~johab/articles/phd.pdf`.