6

In response to your inquiry

Automatic e-mail answer suggestion in a Dutch Contact Centre

Michel Boedeltje and Arjan van Hessen
Telecats and Telecats/University of Twente

Abstract

In the past years, the number of service requests through e-mail has shown an explosive growth, forcing companies and government to set-up contact centres in order to handle these e-mails. Equal to most telephony services handled by call centres, 80% of the incoming e-mails is about 20% of the subjects, making it worthwhile to compose standard answers for at least the 20% most popular questions. Personal answering (each e-mail answered by a human agent) is simply too expensive and not necessary due to this 80-20 rule: well formed predefined answers can cover a significant part of the questions. By using IR and text classification techniques combined with Natural Language Processing, the process of finding the correct answer for a request can be (partly) automated. In this paper we will describe an answer suggestion system using IR based classification and NLP techniques. A practical study using an e-mail corpus of 17,000 incoming e-mails (collected and categorized in a Dutch contact centre), has shown that this approach is able to present the correct answer within a ranked list of 5 possible suggestions, for almost 88% of all incoming e-mails. Furthermore, we will show that this approach can be used as well for spoken content by combining the categorization techniques with the recognition result of the answer on the famous first question: ”Hello, how can we help you?”.

Proceedings of the 18th Meeting of Computational Linguistics in the Netherlands, pp. 85–100
Edited by: Suzan Verberne, Hans van Halteren, Peter-Arno Coppen.
 Contact: michel@telecats.nl

6.1 Introduction

With the ongoing acceptance of e-mail as a fast, cheap and reliable way of communication, companies receive an increasing number of service requests via e-mail. To handle these e-mails, call centres are ”transformed” into so-called contact centres handling both e-mail and telephone calls. Since most service requests cover a relatively small set of problems or questions (the Pareto effect, saying that 80% of the questions is about 20% of the subjects (Reed 2001)), many of these requests may be answered using a relatively small set of standard answers. Handling great amounts of e-mail in a contact centre is a very labour-intensive task, requiring a serious investment of time and money. Automating the answering process could therefore account for serious cost reduction and a significant decrease in response time. Due to the difficulty of automatically selecting the correct answer and thus the risk of sending the incorrect answer to a customer, most companies are reluctant to incorporate such an automatic e-mail answering system in their contact centres. However, automatically suggesting one or more relevant answers to incoming messages provides a good alternative. If we manage to suggest the correct answer in for instance a top-x of relevant answers, the agent only needs to browse through the selected answers and pick the correct one out of these x selected answers, instead of formulating the answer manually or searching the correct answer from all possible answers. Such a system would improve the efficiency in a contact centre and reduce the time spent on answering e-mail. However, one has to tune the number of suggested answers. Suggesting a small number of answers decreases the time an agent has to spend on browsing through the suggestions if the right answer is in these suggestions. However, if the right answer is not present in the suggestions, an agent has to spend extra time on manually searching the correct suggestion. Increasing the number of suggestions increases the chance the right answer is in the suggestions, but also increases the time an agent has to browse all these suggestions. Ideally, no more than 5 suggestions are given to the agents.

6.1.1 Problem statement

The contact centre where this research was performed uses an e-mail management system that enables contact centre agents to handle incoming e-mail efficiently. This system also provides functionality to automatically suggest relevant answers. This answer suggestion routine maps incoming e-mail to a standard question, based on the presence of predefined keywords in the incoming message. Each set of keywords is manually assigned to a standard question and the standard question that has the most keywords in common with the incoming e-mail, links to the best answer suggestion. Although this procedure works fine with a very homogeneous set of e-mails where each suggestion is defined by a well defined set of keywords, it turned out that in real world applications the variety of text in the e-mails is too high, to handle it well with keywords. In the studied case, there was a probability of just 50% that the right answer was present in the 10 best suggestions. So the agent had to browse through 10 suggestions with a chance of only

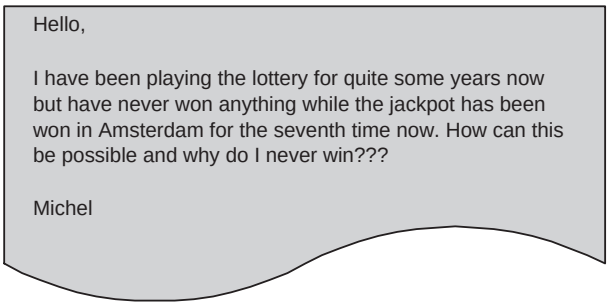

Figure 6.1: Typical e-mail in our contact centre

50% that the correct answer was in the suggestions. As a result, the system became useless. The main goal of this (practical) study was to investigate to what extent Information Retrieval based classification techniques improve the automatic suggestion of answers. In figure 6.1 we printed a typical e-mail (translated to English) for our contact centre of a well-known Dutch lottery.

6.1.2 Classification and speech recognition

Besides suggesting possible answers for written questions (e-mails), we will look at the possibility of using the discussed techniques in speech enabled call routing as well. Instead of confronting a calling customer with a confusing and elaborate IVR[1] menu, we prompt the caller to just say why they call. The spoken utterance is then converted to text by an LVCSR[2] system and classified using the same classification approaches as for e-mail.

6.2 Related work

The concept of using text categorization techniques for assisting agents in answering e-mail is not new. Busemann et al. (2000) developed the ICC mails system to assist call centre agents in answering e-mails by suggesting relevant solutions for incoming e-mail. They use Shallow Text Processing (STP) like word stemming, part-of-speech tagging and sentence types in combination with statistics based machine learning (SML) techniques like neural networks and support vector machines for mapping incoming mail on standard answers. Their experiments on a German e-mail corpus (containing 4,777 e-mails and 74 standard answers of which 47 used in the experiments) showed that the correct answer is selected in about 56% of the incoming e-mails using support vector machines. Neural networks and lazy learners only manage to select the correct standard answers in about 22% to 35% of

[1] IVR (Interactive Voice Response): responding on questions by pressing the buttons on a telephone
[2] Large Vocabulary Continuous Speech Recognition

the cases. Using support vector machines, the correct standard answer is selected within the top 5 results in 78% of the cases.

Gaustad and Bouma (2002) have experimented with an e-mail dataset acquired in a help desk environment in their research on Dutch text classification. Their dataset consisted of 6,000 e-mails, categorized in 69 categories (which have a standard answer assigned to it), but their experiments focused on a subset of 5,518 e-mails categorized in 69 categories. For this dataset, the results ranged from approximately 43% correct for the first suggestion of the system, to 78% correct classification in the best-5 results (the correct answer is present in the first 5 suggestions). For their experiments, Gaustad and Bouma use a Naive Bayes classifier.

6.3 Classification approach

We try to tackle the automatic answer suggestion problem by transforming it into a text classification problem. Each e-mail message is looked at as a document that should be classified and the categories in which they should be classified are the representations of the standard questions. If an e-mail is classified (i.e. mapped to a standard question), we can simply suggest the answer that is associated with the standard question representing the category. The classification of new messages is done by determining the similarity between the new messages and previously answered messages. Based on the assumption that similar questions require similar answers, the new message can then be categorized in the category that stores previously answered messages that are most similar to the new message. We state that automatically determining the similarity between new messages and previously answered messages using IR techniques outperforms the basic classification approach using manually determined keywords.

We have developed an e-mail answer suggestion system in which two classification routines can be used. We incorporated a profile based classification routine called the Rocchio classifier and an example based classification routine called the K-Nearest-Neighbour classifier. In this system, each classifier can be used using either the TF.IDF (Salton and McGill 1983) or Okapi (Robertson and Sparck Jones 1997) relevance weighting scheme.

6.3.1 Rocchio classifier

A profile based classifier is basically a classifier which embodies an explicit, or declarative, representation of the category on which it needs to take decisions. Rocchio developed an algorithm for relevance feedback for use in the vector space information retrieval model, which can be adapted to serve as a profile-based classifier. Joachims (1997) describes the use of the Rocchio Classifier using TF.IDF weights, but other weighting schemes may also be used. In the training phase, the classifier learns to classify documents by calculating a prototype vector $\overrightarrow{c_j}$ for each class C_j. In this training phase, both the normalized vectors of the positive examples for a class as well as the negative examples for a class are used. Each prototype vector is calculated as a weighted difference of the positive and negative

examples:

$$\vec{c}_j = \alpha \left(\frac{1}{|C_j|} \sum_{\vec{d}_j \in C_j} \frac{\vec{d}_j}{||\vec{d}_j||} \right) - \beta \left(\frac{1}{|D - C_j|} \sum_{\vec{d}_j \in D - C_j} \frac{\vec{d}_j}{||\vec{d}_j||} \right)$$

Where C_j is the set of training documents assigned to class j and $||\overrightarrow{d_j}||$ denotes the Euclidean length of a vector $\overrightarrow{d_j}$. Additionally, α and β are parameters that adjust the relative impact of positive and negative training examples, recommended to be 16 and 4 respectively. However, in this study the optimal parameter settings for α and β are 1 and 8 respectively, implying that the influence of negative examples should be 8 times as big as the positive examples for the best classification results. The resulting set of prototype vectors (one for each class) represents the learned model that can be used to classify a new document d' using:

$$H_{TFIDF}(d') = \arg\max_{C_j \in C} cos(\vec{c_j}, \overrightarrow{d'})$$

The classification function H_{TFIDF} (H for hypothesis) returns the category that has the highest similarity score (using the cosine function, but other similarity functions may also be used) with respect to the document to be classified. This approach can be slightly adjusted to return a ranked list (in decreasing order of similarity) of categories that are suitable for document $\overrightarrow{d_j}$ by ignoring the $\arg\max$ function and ordering the calculated similarity scores for each category in descending order (cut off at a certain threshold if pleased).

6.3.2 K-Nearest-Neighbour classifier

Example based classifiers do not build a representation for each category and do not involve in a true training phase (these classifiers are also lazy learners). A commonly used algorithm for example-based classification is the K-NN (K-Nearest-Neighbour) algorithm, implemented by Yang (1994) in the Expert System. The conditional probability that a document d_j is classified in category c_k by human judgement is given by:

$$Pr(c_k|d_j) \approx \frac{\#(assign(c_k, d_j))}{\#(d_j \in D)}$$

Where $d_1, ..., d_m$ are unique training documents and $C_1, .., C_l$ are unique categories. Furthermore, $\#(assign(c_k, d_j))$ is the number of times category c_k is assigned to document d_j and $\#(d_j \in D)$ is the number of times document d_j occurs in the document collection D. This probability is calculated since a document may have more than one occurrence in the training sample (at least after text normalization like stopword removal and stemming). Usually this equation results in a 0 or 1, indicating a category is or is not assigned to a document. The relevance score is then calculated by comparing the query q to the first K documents $d_j \in D$ using a similarity measure like the inner product or cosine, and multiplying the result with the conditional probability calculated earlier:

$$rel(c_k|q) \approx \sum_{j=0}^{K} sim(q, d_j) \times Pr(c_k|d_j)$$

Where $sim(q|D_j)$ is the similarity score calculated by the IR component and both $sim(q|d_j)$ and $rel(c_k|q)$ are scores, not probabilities. The results can be used to return a ranking (in descending order of relevance) of categories most suitable for the new document. Again, this ranking can be cut off at a certain threshold.

6.4 Natural Language Processing

E-mails have a "lower" status than traditional letters and therefore are often written sloppily: they contain a lot of spelling errors and grammatical incorrect sentences. This increases the number of words used and therefore may negatively influence the performance of the classification algorithms. To overcome (most part of) this problem, we use basic Natural Language Processing to "normalise" the e-mail before using its contents in the classification algorithms.

6.4.1 Lexical normalisation and stopword removal

The first (and most basic) step is to remove stopwords and apply some lexical normalisation to each e-mail. Lexical normalisation is nothing more than a simple process of removing all unwanted characters and strings like the sender's e-mail address or postal code and removing diacritics. Stopword removal is applied to reduce feature size and speed-up the indexing and classification process.

6.4.2 Stemming

By applying stemming we hope to improve the classification process by reducing the morphological variance of terms. If a set of documents are all about the same topic (or pose the same question in our problem), but use different morphological variants (like *swimming, swum, swam* and *swim*), a classification method is unable to relate the documents based on these terms. If we apply stemming, all documents from this set now contain the same morphological variant (i.e. *swim*) and therefore can be related. In this study we use a dictionary based stemming routine provided in the Lingware tool-kit[3]. If a word could not be found in the dictionary, the stemmer uses similar words (i.e. with the same ending and word class) for which the stemming procedure is known, and applies the same procedure to the unknown words.

6.4.3 Decompounding

Decompounding (or compound splitting) is a specific NLP routine often very useful for compounding languages like Dutch, German or Finish. By decompounding we intend to improve the classification accuracy by improving the precision and recall of the IR component of the classification system. Chen (2002) showed that decompounding can improve both recall and precision in Dutch and German IR systems. Also, Monz and De Rijke (2001) have performed successful experiments

[3]provided and implemented by Carp Technologies

on using decompounding in a Dutch IR system which caused an increase in average precision of 6.1%. Like Monz and De Rijke (2001) the Dutch lexicon of Celex is used to implement a compound splitter in our e-mail classification system.

6.4.4 Part-of-Speech tagging

Part-of-Speech (POS) tagging has proven to be very useful in IR and text categorization, mostly due to its use for disambiguation of terms. In our e-mail classification system we use POS tagging for disambiguation of terms before stemming them and as a feature selection mechanism. For instance, words from so-called open word classes carry more meaning than words from closed word classes. Kraaij and Pohlmann (1996) stated that the majority of successful query terms for an IR system in a collection of newspapers are nouns (58%), followed by verbs (29%) and adjectives (13%), while other categories are negligable. In our system we use an unsupervised transformation based tagger (provided in the Lingware toolkit).

6.4.5 Spelling correction

E-mails may contain (many) spelling errors and typo's which (for similar reasons as stemming) does not help in retrieving and classifying an e-mail. To correct (the majority of) spelling errors and typo's in our e-mails, we use a context based spelling correction routine from the Lingware toolkit, based on N-grams, Levenhstein distance and models of common made typing errors (Jurafsky and Martin 2000).

6.5 E-mail experiments

For this practical study the contact centre in question has provided a set of approximately 30,000 e-mails. Unfortunately, this corpus has not been constructed carefully for classification purposes. After removing "nonsense" e-mails (like spam, empty e-mails, error messages, etc.) and disambiguation of the categories a corpus of 16,798 e-mails categorized in 37 categories remains. The average number of e-mails per category is 454, the largest category contains 3,593 e-mails and the smallest one contains 106 e-mails. Figure 6.2 shows the distribution of e-mails per category. The results of the classification experiments are expressed in best-x classification accuracy. If the system suggests the correct answer suggestion within the top 5 of suggestions for 50% of the e-mails, the best-5 classification accuracy is 50%. We chose this best-x classification accuracy over i.e. mean reciprocal rank (MRR) to improve readability for the stakeholders of this study (contact centre managers). For comparison, we will also denote the MRR in the results section.

This study focussed on the use of IR based classification systems for suggesting relevant answers in a contact centre and Natural Language Processing to improve the classification accuracy of these systems. We conducted two series of experiments, the first series focuses on the selected classification approaches (without

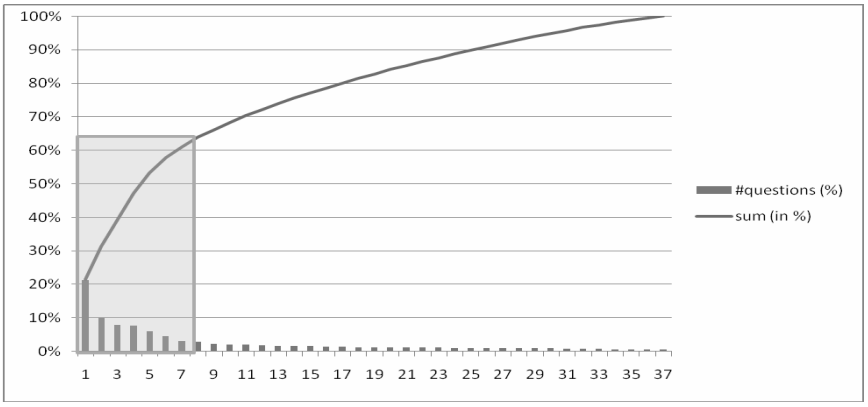

Figure 6.2: Distribution of the number of e-mails over the set of 37 categories. The largest category contains 3,593 (21%) messages, the smallest 106 (0.63%). Total amount of messages is 16798. The grey line shows the summed messages as a percentage (from 21% to 100%). The square shows the Pareto effect: the first 8 questions (21.6%) are responsible for 65% of the total amount of questions.

NLP), as the second series focuses on the use of NLP in these classification approaches. All experiments are performed using 5-fold cross validation. In table 6.5 we listed the parameter settings for our classification models. As mentioned before in section 6.3.1 the optimal parameters ($\alpha = 1$ and $\beta = 8$) for manipulating the relative influence of positive and negative examples for the Rocchio classifier differ significantly from the default parameters ($\alpha = 16$ and $\beta = 4$). The nature of the Rocchio classifier is to determine an optimal margin between the centroids of categories (the prototype vectors). The optimal margin can be found by parameter estimation that works as a feature selection procedure to find those features that are most relevant to distinguish a category from the other categories (Moschitti 2003). By increasing the influence of the negative examples with respect to the positive examples, we smoothly remove features from the positive examples that are irrelevant for distinguishing this category from the others. The relatively high influence of the negative examples in this study, shows that the set of distinguishing features for each category is relatively small and that the feature sets of the e-mail messages have an above average overlap. This means that the majority of the words used in the email messages sent to the callcenter is quite similar.

6.5.1 Experiments

In the first series of experiments we have tested two classification systems: The example based classifier (K-Nearest-Neighbour) and the profile based classifier (Rocchio). These classifiers are tested using two term relevance weighting

Table 6.1: Parameter settings for the classification models

K-NN classifier	$K = 50$
Rocchio classifier	$\alpha = 1$ and $\beta = 8$
Okapi weighting scheme	$b = 0.75$ and $k = 2$

schemes: The TF.IDF weighting scheme and the Okapi weighting scheme. Both classification approaches can use either the cosine similarity measure or the inner product similarity measure.

The second series of experiments focussed on the use of NLP within our classification approaches. We have experimented with the NLP techniques discussed in section 6.4 apart and the combination of several of these techniques together to determine the most optimal system implementation for this specific problem. In our study we found that not all NLP techniques improved the classification accuracies of both methods. For instance, decompounding caused a significant increase in accuracy for the example based classifier using Okapi weights and inner product similarity but caused a significant decrease in accuracy for the profile based classifier. In general, the example based classifier benefited more from the NLP routines than the profile based classifier.

The main results of our experiments are listed in table 6.5.1 and plotted in figure 6.3. In table 6.5.1 we printed the Mean Reciprocal Rank (MRR) and denoted the best-x classification accuracy for $x \in \{1, 3, 5\}$. In the e-mail answer suggestion system we are specifically interested in the best-5 performance, since trained contact centre agents are capable of overseeing 5 possible answers in one glance. To prove our hypothesis that de IR based classification approaches outperform the manually determined keywords based approach, we also printed the results of this keyword based approach. As a reference, the best-guess method "classifies" documents by simply suggesting the answer linked to the largest category first, the second largest second, etc.. Besides the results of our best performing classifiers, we also included the results of our best-performing classifiers combined with NLP techniques. Without NLP the example based classifier has a MRR of 0.65 and a best-5 accuracy of 84.2%, whereas the profile based classifier has a MRR of 0.61 and best-5 accuracy of 77.4%. Compared to the keywords-based approach, this is a major improvement. In the table and figure we can also see that applying NLP techniques improves the classification accuracy of the example based classifier (and the MRR increases to 0.71 due to the increase in best-1 classification accuracy), whereas the profile based classifier does not benefit that much from using NLP. Tables 6.5.1 and 6.5.1 show more detailed results of the classification experiments using NLP techniques. The example based classifier benefits most from these techniques, except from the POS-tagging/Stemming combination which causes a decrease of 10 to 12 percentage points in best-3 and best-5 classification accuracy. Using the same NLP techniques in the profile based classifier does not show any significant improvement, but rather worsened the performance

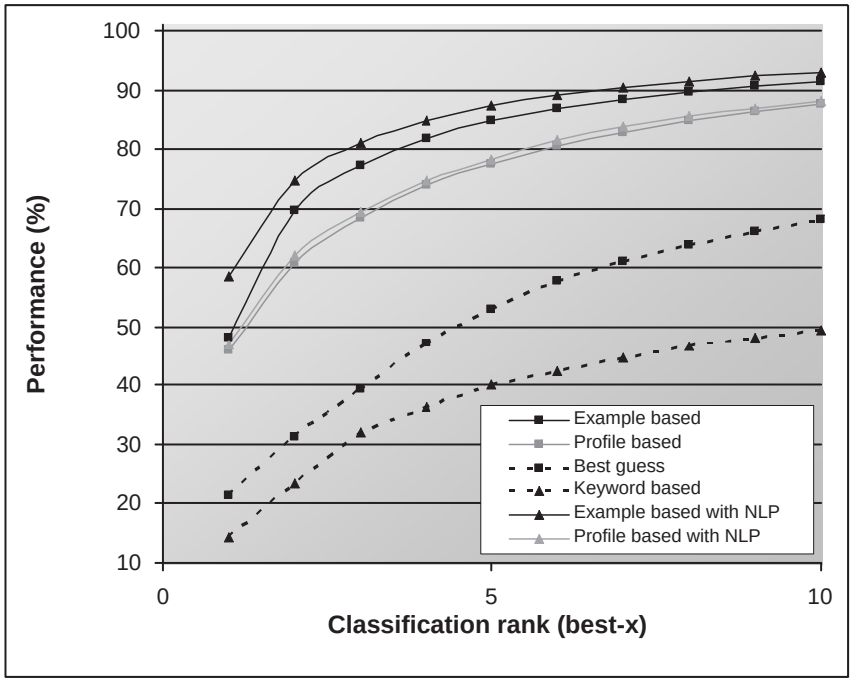

Figure 6.3: Overview of the best-x classification accuracies for the best performing example and profile based models, compared to the same models combined with NLP techniques. For comparison, the accuracy of both the best-guess approach and keywords based approach is also plotted.

of the profile based classifier (especially when decompounding was applied). Our best classification approach (Example based with NLP) is able to suggest the correct answer within a set of 5 suggestions for 87.4% of the incoming e-mails (with a MRR of 0.71).

6.6 Speech enabled call routing

As previously mentioned we also use these classification approaches in a speech enabled call routing system. Applications of speech based routing (i.e. *How may I help you?* from Gorin et al. (1997)) where a caller poses his question to a computer and is routed to for instance one of the five possible departments are widely known and implemented.

In our speech enabled call routing application, we pose a similar question and try to map de question to one of our standard questions to determine the correct action. These actions vary from playing a prompt with the most likely answer to

Table 6.2: Classification accuracies of the baseline experiments. The table lists the Mean Reciprocal Rank and best-x classification accuracies for $x \in \{1, 3, 5\}$

Approach	**MRR**	**Best-1**	**Best-3**	**Best-5**
Example based	0.65	48.2%	77.2%	84.8%
Profile based	0.61	45.9%	68.4%	77.4%
Example based with NLP	0.71	58.5%	81.0%	87.4%
Profile based with NLP	0.62	47.0%	69.5%	78.3%
Best-guess	0.36	21.4%	39.3%	53.1%
Keyword-based	0.26	14.2%	31.9%	40.0%

Table 6.3: Classification accuracies of a selection of the NLP experiments using the example based classifier. The table lists the increase or decrease in best-x classification accuracies for $x \in \{1, 3, 5\}$

NLP technique	**Best-1**	**Best-3**	**Best-5**
Example based, baseline	52.07%	74.63%	82.15%
Stopword removal	+3.58	+3.1	+2.62
Decompounding	+6.48	+6.42	+5.25
Stemming	+5.81	+5.96	+4.73
POS-tagging	+6.24	+6.01	+4.68
POS-tagging/Stemming	+3.99	-10.36	-12.79
Stopwords/Decompounding	+4.39	+4.73	+3.78
Decompounding/Stemming	+5.19	+6.24	+4.91
Stopwords/Stemming	+7.16	+6.84	-1.41
Decompounding/POS-tagging	+5.93	+6.11	+4.78
Stopwords/Decompounding/Stemming	+4.05	+4.26	+3.69

Table 6.4: Classification accuracies of a selection of the NLP experiments using the profile based classifier. The table lists the increase or decrease in best-x classification accuracies for $x \in \{1, 3, 5\}$

NLP technique	**Best-1**	**Best-3**	**Best-5**
Profile based, baseline	45.87%	68.45%	77.40%
Stopword removal	-1.77	-0.78	-0.02
Decompounding	-4.11	-3.72	-2.69
Stemming	+0.36	-0.24	-0.56
POS-tagging	+1.16	+1.06	+0.91
POS-tagging/Stemming	-2.28	-1.85	-1.55
Stopwords/Decompounding	-5.96	-5.01	-3.87
Decompounding/Stemming	-3.41	-3.41	-3.34
Stopwords/Stemming	-3.36	-0.664	-0.05
Decompounding/POS-tagging	-3.36	-2.33	-1.58
Stopwords/Decompounding/Stemming	-7.19	-5.06	-3.86

routing to a self-service application or specific department of the company. This type of application brings an extra speech recognition task to the application. The LVCSR results of each spoken utterance are sent to the classification system and a ranked list with best matching standard questions is returned. The caller can then chose the best matching standard question to proceed in the application.

6.6.1 Corpus

To determine the classification accuracies of these classification systems if instead of written text, recognized speech is used, we have collected a set of 3,322 spoken utterances in our speech enabled call routing application implemented in a Dutch contact centre of a telecom provider. The recorded utterances can be categorized in 36 categories with an average category size of 92. The smallest category contains only 3 utterances as the largest category contains 279 utterances. The size of this corpus is a bit small for the intended experiments, but since all utterances have to be manually transcribed (in order to study the effect of speech recognition) and categorized for training and testing, no more data was available at the current time.

6.6.2 Experiments

The experimental set-up is similar to that of the e-mail answer suggestion experiments. We focus on the example based classification approach (K-Nearest-Neighbour with $K = 25$) with the Okapi weighting scheme ($b = 0.75$ and $k = 2$) using inner product similarity and no additional NLP techniques other than stopwordremoval. For the LVCSR we use a commercially available speech recognizer designed for dictation. Within this recognizer we use the 'Unisex' acoustic model

Table 6.5: Classification accuracies of the speech recognition classification experiments. The table lists the Word Error Rate, Mean Reciprocal Rank and best-x classification accuracies for $x \in \{1, 3, 5\}$

Approach	**WER**	**MRR**	**Best-1**	**Best-3**	**Best-5**
Transcriptions	0%	0.79	69.5%	87.6%	91.9%
General context	74%	0.48	38.1%	53.3%	59.1%
Website context	67%	0.58	48.0%	64.5%	70.1%
Transcription context	55%	0.67	57.1%	75.0%	80.5%
Best-guess	-	0.21	8.4%	20.1%	30.0%

because the caller identity and gender are unknown (before we recognize the utterance). We perform experiments with four different context models, each trained with a specific type of documents:

- A general context with additional training of CGN data (telephone and face-to-face conversations (Oostdijk 1999)): 'General context'
- The context above, additionally trained with relevant context information of the telecom provider (taken from the company's website: 'Website context'
- The context above, additionally trained with transcriptions of spoken utterances in the speech enabled IVR application: 'Transcription context'

For comparison, we perform our experiments by classifying the speech recognized texts and also the orthographic transcriptions to study the influence of the speech recognition induced errors on our classification accuracy.

The results of our experiments are listed in table 6.6.2 and plotted in figure 6.4. For each of the experiments we denoted the MRR, best-x classification accuracy for $x \in \{1, 3, 5\}$ to reflect the classification performance and the Word Error Rate to reflect the speech recognition performance.

The WER (Word Error Rate) of the recognized utterance is pretty high, mostly because we are forced to use an unisex acoustic model and have to deal with noisy telephone speech. However, if we apply a well trained context (language model), the WER decreases to 55% and a best-5 classification accuracy increases to over 80%. Unfortunately, this is 10% less than the accuracy we could have yielded if there were no speech recognition induced errors. The next goal would be to improve the speech recognition component in order to decrease the WER; this can be done by adjusting the language models with more transcriptions. Moreover, if we manage to incorporate a gender detection routine, we can apply gender-specific acoustic models in the speech recognition task. The classification accuracies are also expected to increase if we expand the size of the training set: In these series of experiments we were forced to use just 3,300 examples instead of the almost 17,000 examples for the e-mail experiments, while the number of categories are almost equal (36 for speech opposed to 37 for e-mail).

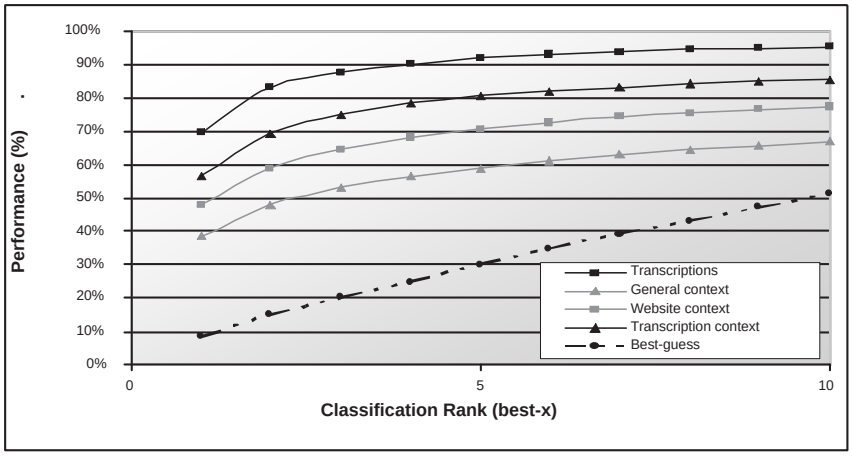

Figure 6.4: Overview of the best-x classification accuracies for the various context models (General, Website, Transcription) and full hand made transcription. For comparison, the accuracy of the best-guess approach is also plotted. The figures are based on the spoken Telecom corpus.

6.7 Conclusions and future work

In our introduction we stated that IR based classification would outperform the keyword based classification approach in our e-mail answer suggestion problem and that the use of Natural Language Processing would even further improve the accuracy. We showed that by using IR based classification approaches, the best-5 classification accuracy more than doubled from 40% to approximately 85%, meaning that for almost 85% of the incoming e-mails, the correct answer suggestion is listed within a ranked list of 5 possible answer suggestions. If we apply Natural Language Processing within the classification task, the best-5 classification accuracy rises to more than 87%. A relatively small increase, but if we focus on the best-1 classification accuracy, the increase is more than 10%. In conclusion we developed an e-mail answer suggestion system that suggests the correct answers within a list of 5 possible suggestions in 87% of the times and, moreover, places the correct answer suggestion at the top of this list in almost 60% of the cases. Furthermore we showed that these classification approaches are also well suitable in speech enabled call routing systems where callers respond on the question: ”How may we help you?”.

Our future work will focus on the improvement of the speech enabled call routing applications. We intend to boost the best-3 classification accuracy over 80% by improving the speech recognition results and better matching of these results with the classification models. To improve the classification accuracy, we may also ben-

efit from NLP techniques to find a better match between the classification models and the speech recognized utterances (e.g. by using feature expansion by adding frequent confusion words). In order to provide reliable confidence information to the classification results, we will also focus on the use of other classification approaches. If we are confident that a standard question is the best match for the spoken utterance of the caller, we can improve the self-service level by providing the correct answer immediately instead of prompting the caller to select his question from a set of relevant questions.

References

Busemann, S., S. Schmeier, and R.G. Arens (2000), Message classification in the call center, *Proceedings of the sixth conference on Applied natural language processing*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 158–165.

Chen, A. (2002), Cross-language retrieval experiments at clef 2002, *Proceedings of CLEF-2002*, pp. 28–48.

Gaustad, T. and G. Bouma (2002), Accurate stemming of dutch for text classification, *Language and Computers* **45** (1), pp. 104–107.

Gorin, A. L., G. Riccardi, and J. H. Wright (1997), How may I help you?, *Speech Communication* **23** (1/2), pp. 113–127.

Joachims, T. (1997), A probabilistic analysis of the rocchio algorithm with tfidf for text categorization, *in* Fisher, Douglas H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, Nashville, US.

Jurafsky, D. and J.H. Martin (2000), *Speech and Language Processing*, Prentice Hall, New Jersey.

Kraaij, W. and R. Pohlmann (1996), Viewing stemming as recall enhancement, *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, New York, USA, pp. 40–48.

Monz, C. and M. De Rijke (2001), Shallow morphological analysis in monolingual information retrieval for dutch, german and italian, *Proceedings CLEF 2001*, Springer Verlag, pp. 262–277.

Moschitti, A. (2003), A Study on Optimal Parameter Tuning for Rocchio Text Classifier, *LECTURE NOTES IN COMPUTER SCIENCE* pp. 420–435, Springer.

Oostdijk, Nelleke (1999), Building a corpus of spoken dutch, *Computational Linguistics in the Netherlands 1999, Selected Papers from the Tenth CLIN Meeting, December 10, OTS Utrecht*.

Reed, William J. (2001), The pareto, zipf and other power laws, *Economics Letters* **74** (1), pp. 15–19.

Robertson, S.E. and K. Sparck Jones (1997), Simple, proven approaches to text

retrieval, *Technical report*, City University London and University of Cambridge.

Salton, G. and M.J. McGill (1983), *An introduction to Modern Information Retrieval*, McGraw-Hill.

Yang, Y. (1994), Expert network: Effective and efficient learning from human decisions in text categorisation and retrieval, *in* Croft, Bruce W. and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag, pp. 13–22.