Information retrieval based call classification

Jan Kneissler, Anne K. Kienappel, Dietrich Klakow

Philips Research Laboratories
Weißhausstraße 2, D-52066 Aachen, Germany
{Jan.Kneissler Anne.Katrin.Kienappel Dietrich.Klakow}@philips.com

Abstract

In this paper we describe a fully automatic call classification system for customer service selection. Call classification is based on one customer utterance following a "How may I help you" prompt. In particular, we introduce two new elements to our information retrieval based call classifier, which significantly improve the classification accuracy: the use of a-priory term relevance based on class information, and classification confidence estimation. We describe the spontaneous speech recognizer as well as the classifier and investigate correlations between speech recognition and call classification accuracy.

1. Introduction

Flexible, spontaneous speech interfaces have long been a popular idea in the context of telecom applications. However, such interfaces require both an advanced level of speech recognition and speech understanding technology.

This paper is concerned with automatic call classification, a task that has to be completed in e.g. automatic call steering applications. In such applications it is desirable to allow the customer to use natural language. Based on a spontaneous customer utterance, the system must be able to reliably and fully automatically assign most calls to the correct service. We present in this paper a system with a call classifier that uses information retrieval technology.

The paper is organized as follows. After a brief introduction of the target database and application, we describe the information retrieval based call classifier including two novel extensions: a-priory term relevance due to class information and classification confidence estimation. We then go on to describe our experimental set-up including the speech recognizer. We finally report results from speech recognition and call classification experiments.

2. The OASIS call classifier task

We present an automatic call classification system for telecom general inquiries based on the OASIS database collected by BTexact Technologies (see [1], [2], [3]). The data consists of recordings and transcriptions of the first customer utterance in operator assistance calls to BT, which are to be recognized and assigned to one of a set of 19 classes representing different services that the callers should be re-directed to.

We use the first 1k utterances (containing 23k words) in the database for testing. A further 7.5k utterances are available for training and development of acoustic models, language models and the call classifier.

For training of the recognizer, we use two additional UK English spontaneous speech databases, ServiceA and ServiceB, which have also been collected by BT. They are both of similar size and also consist of first customer utterances from calls to different telecom customer services and match the OASIS database very well acoustically.

3. Call classification: Theory

3.1. Classification via info retrieval

In classical information retrieval (see e.g. [4]) a large document collection is searched for specific information by entering a – typically short – query. Usually the reply to a query is given by a list of documents that are ranked with respect to decreasing relevance. One well-known example for this scenario are search engines for the world wide web.

However, it is quite a natural idea to apply info retrieval to text categorization purposes. The recipe to do that in the framework of call classification is simple (e.g. [5]):

- training material → document collection
- evaluation material → queries
- service assigned to retrieved calls → output of classifier

The reasoning behind this approach is that the service assigned to the most resembling utterance out of the training set has highest chances to be also the correct choice for a given test utterance. So, the fact that information retrieval approaches estimate *relevance* by text similarity measures – which in general is only an approximation – is helpful in our context. However, the recipe leaves a few questions open:

- Should all the training data belonging to the same service be gathered into one document for each service?
- Should one take only the best matching document into account or also retrieved documents with lower ranks?

We observed that clustering calls according to the services severely degrades classification performance, and did no further experiments in that direction (see [6] for a detailed investigation of this effect). The second question is discussed in more detail in section 3.5.

3.2. The vector space modeling approach to IR

The vector space modeling paradigm (see e.g. [7]) implies that documents and queries are represented by k dimensional vectors whose components correspond to the occurrence counts of terms (words or word stems). This is sometimes formulated in a sloppy way by saying that documents and queries are treated as *bags of terms*, i.e. their ordering in the text is ignored.

The vector components of a document d are typically constructed as a monotonically increasing function f of the term counts:

$$d_t \quad := \quad f(\text{\# occurrences of term } t \text{ in document } d), \qquad (1)$$

and correspondingly for queries.

A very simple example for a similarity function is the angle (or equivalently its cosine) between the document and query vector: $\cos\phi(\mathbf{d},\mathbf{q}) = \frac{\mathbf{d}\cdot\mathbf{q}}{|\mathbf{d}|\,|\mathbf{q}|}$. A very popular generalization is to replace the inner product by a general scalar product $\langle\mathbf{d}|\mathbf{q}\rangle = \sum_{i,j} g_{ij} d_i q_j$, usually taking a diagonal metrics $g_{ij} = \delta_{ij} r(i)$. The values on the diagonal are usually defined by $r(t) := -\ln(n(t)/n_{\text{doc}})$, where $n(t)$ denotes the number of documents in the database that contain the term t and n_{doc} is the total number of documents (*inverse document frequency*). This results in the following similarity measure:

$$S(\mathbf{d},\mathbf{q}) \;=\; \sum_{\text{terms } t} \frac{d_t}{|\mathbf{d}|}\,\frac{q_t}{|\mathbf{q}|}\,\ln\frac{n_{\text{doc}}}{n(t)} \qquad (2)$$

3.3. Normalization by self-scores

In a call steering system, there may be the wish to handle only a certain fraction of the incoming calls automatically. In order to filter out the calls that are more likely to be correctly classified, a comparison of the retrieval scores for different queries is needed. But the similarity measure of equation (2) is not suited for a direct comparison since it heavily depends on the query and the class. We use the following method to compensate this effect:

Self-Score Normalization: Given a similarity function S, construct a new one S^* (depending on a parameter $\mu \in [0,1]$), by

$$S^*(d,q) \;:=\; S(d,q) - (1-\mu)\,S(d,d) - \mu\,S(q,q) \qquad (3)$$

If the old similarity function S satisfies

$$\max_x S(d,x) = S(d,q) \;\Leftrightarrow\; d = q \;\Leftrightarrow\; \max_x S(x,q) = S(d,q),$$

then the new S^* has nice properties: it takes only values ≤ 0 and $S^*(d,q) = 0$ if and only if there is an exact match between document and query[1].

3.4. A-priori term relevance due to class information

The third factor (inverse document frequency) in the classical retrieval vector space based formula (2) can be regarded as an estimate for the term's relevance (terms occurring in many documents have less discrimination potential). It neglects a very important piece of information namely the partition of the training document set according to the class labels assigned to them.

A term's importance should not be decided by the simple rule "the less documents it occurs in, the more relevant it is". Moreover, there are two, somehow opposing criteria:

a) how many documents of a specific class contain the term (the more, the better),

b) within how many classes exist documents that contain the term (the less, the better).

In other studies on IR based classification, both principles have been reconciled by putting all training documents for a class into one big document as proposed in the first question of section 3.1. Thus terms count weighting and inverse document frequency weighting and are expected to somehow accomplish the effects a) and b) automatically. But there are other difficulties with that approach; generally speaking, comparing something to a bunch of examples individually (picking out the one that matches best) works better than comparing it to the unstructured union of examples.

In our classification experiments, we modeled term relevances more explicitly according to the principles a) and b), by saying that a term is relevant if it is *class typical* (there is at least one class for which the term is very frequent) and it is *class specific* (there are not many classes for which the term occurs). The two properties are modeled by individual factors in the second row of the following similarity measure:[2]

$$\begin{aligned} S(\mathbf{d},\mathbf{q}) &= \sum_{\text{terms } t} d_t\; q_t\; r(t) \\ r(t) &= \left(\max_c \tfrac{n_{\text{doc}}(c,t)}{n_{\text{doc}}(c)}\right) \cdot h\!\left(\tfrac{n_{\text{class}}(t)}{n_{\text{class}}}\right) \qquad (4) \\ h(x) &= 1 - \sqrt{2x - x^2}. \end{aligned}$$

Maximization in $r(t)$ runs over all classes c and

$n_{\text{doc}}(c,t)$**:** number of documents in the class c containing t,

$n_{\text{doc}}(c)$**:** number of documents in class c,

$n_{\text{class}}(t)$**:** number of classes in which there is a document containing t,

n_{class}**:** total number of classes.

The plot of the function h is a quarter-circle with zero slope at $x = 1$ and infinite slope at $x = 0$. This function performed best out of a set of several functions we have tried.

3.5. Voting and classification confidence estimation

In section 3.1 we raised the question of using more of the information in the retrieval ranking can than just the best score.

A simple answer is N-best voting: the classifier's output is given by the class that obtains the most votes from the N most relevant documents. The ranking scores may also be included into the voting process.

Going one step further, we used N-best voting in combination with an estimation of classification confidences. To describe the method, let us first introduce a little bit of terminology:

For a specific service c and a ranked list L of documents (as produced by a query) let there be a point on the 2-dimensional Euclidean plane defined by the coordinates

$$\begin{aligned} x &:= \text{score in } L \text{ of first document belonging to class } c \\ y &:= \text{score in } L \text{ of first document } \textit{not} \text{ belonging to } c \end{aligned}$$

We denote the point (x,y) by *representation* of L in the *confidence plane* for class c. In other words, on a confidence plane we plot the score for the hypothesis c against its strongest rival's score.

We are now focusing on one fixed class c and will give an estimate for the confidence for c of a ranking list L, depending on the position of L's representation. To this purpose we use a leaving-one-out rankings of the training material as follows:

- For every training utterance produce a ranking of all other training utterances.
- Color the corresponding point on the confidence plane green if the training utterance belongs to class c, color it red otherwise.

[1] For the cosine similarity measure of equation (3.2) this condition on S is true (in the bag-of-words sense, i.e. $d = q$ should be read as "d and q have identical bag-of-words representations").

[2] Note the other difference to eq. (2): no vector length normalization.

Thus we get $n_{\mathrm{doc}}(c)$ red points and $n_{\mathrm{doc}} - n_{\mathrm{doc}}(c)$ green data points.

The idea of the confidence estimation is: for ranking L, the confidence for the hypothesis that class c is the correct one is given by the density of green points on the confidence plane of c at the place of the representation of L. The problem is how to estimate this density on basis of the finite set of red and green points corresponding to the training set. Obviously, counting the green points in a sample set of points "around" the representation of L should give a good approximation; but how large should we choose the sample set? There is a trade off between incorrectness due to statistical variation (small sample set) and deviation due to contamination with points that are too far away (large sample set). We decided to optimize the sample set size n in a predefined range (maximum at 5% of training material size) and use the expectation value lowered by two standard deviations (i.e. at a confidence level of 97%) as target function.

4. The speech recognizer

4.1. Acoustic models

Our acoustic models are continuous Gaussian mixture density hidden Markov models. They are trained using a Viterbi algorithm. The feature vectors consist of 31 linear discriminant analysis (LDA) feature components and are computed with a 10 ms frame shift. The Gaussian densities have diagonal variances, which can be tied and thus shared between densities.

The entire acoustic training data contains to $\approx$ 36 hours of speech, 25 of which are from the OASIS and ServiceA databases and 11 of which are from the ServiceB database. The data is distributed almost equally between male and female speakers. Most training and test data was not gender-labeled. We used a single Gaussian mixture model (estimated iteratively on the training corpus) per gender for gender classification of unlabelled utterances. Whenever gender dependent acoustic models are used, the LDA matrix is also gender dependent.

We use context dependent phoneme models, and decision trees to determine state tying. We found that a triphone context is sufficient for this task; an extension to quinphone context yielded no significant improvements.

4.2. Vocabulary

We use two recognizer vocabularies, of 4k and 5k words. The 4k vocabulary consists of all the words that occur more than once in the OASIS and ServiceA training text. This vocabulary was extended to 5k using additional words from a 64k background lexicon by training an optimized unigram language model for the background lexicon and by picking the most likely words accordingly. This unigram language model was a linear interpolation of models trained on various corpora and using different smoothing techniques. Some of the corpora where automatically selected sub-corpora of ServiceB and corpora from LDC and ELRA.

Whenever (unigram) weights were associated with the pronunciation variants in the recognition lexicon, they were based on the counts of the best-scoring variants in the training data. For words that do not occur in the training data, the probabilities are equally distributed over all pronunciation variants.

After the initial set of experiments, word phrases where added to the vocabulary. The words and phrases are merged to longer units based on a frequency-criterion on the training corpus. After each merging, step the frequencies are recalculated. The length of phrases can grow with iterations (e.g. "hello_i_wonder_if_you_can_help_me"). To avoid phrases that do not match the domain, only OASIS was used as the training set. The final number of 300 phrases optimizes the perplexity on the development data for a bigram language model.

vocabulary	4k	4k	5k
OOV on test set	1.4%	1.4%	1.0%
no. of word phrases	0	300	300
LM-range	bigram	trigram	trigram
Smoothing	mrg. b.-o.	mrg b.-o.	fully opt.
PP	36.5	32.2	24.9
no. of mixtures	1k	2×2.5k	2×4k
no. of Gaussians	59k	2×72k	2×116k
no. of variances	1	2×1	2×116k
gender dependent	no	yes	yes
hrs. train. speech	26	26	37
pron. var. weights	no	yes	yes
phonetic context	WW	WW	XW
WER	48.3%	38.5%	33.0%

Table 1: Recognizer specifications. OOV stands for out-of-vocabulary, PP for perplexity,WW for within-word context modeling and XW for position dependent across-word context modeling, WER for word error rate. Note that the test set perplexity is evaluated on word (not phrase) level.

4.3. Language models

In the reported experiments we use three different language models. The main training text was the transcription of the OASIS training data. The standard training was performed using the marginal backing off strategy. A fully optimized language model was compiled for the 5k vocabulary using using additional corpora from LDC and ELRA and ServiceB transcriptions as additional training texts. Also all discounting parameters (15 per component language model) were optimized on a development subset of the training data. For additional smoothing a linear interpolation of the optimized word models and the class models was done. The test set perplexities of the different language models are listed in table 1.

4.4. Recognizer performance

In table 1 we list the most important specifications of the three recognizers used to produce the texts for our classification experiments (see figure 3). The total WER reduction by 15.3% absolute from the worst to the best system is due to many small improvements (between absolute 0.5% and 3.0% WER reduction) achieved with the changes indicated in the rows of table1.

5. Results

5.1. Classification confidence

To evaluate the classification confidence measure, all test utterances have been distributed into 10 groups according to their estimated classification confidence. Over 40% of the test material fell into the highest confidence class (interval [0.9,1.0]). The precision has been measured individually within the individual confidence level groups. The result is shown in figure 1. The fact that the curves are close to the diagonal indicates that our classification confidence measure produced realistic values.

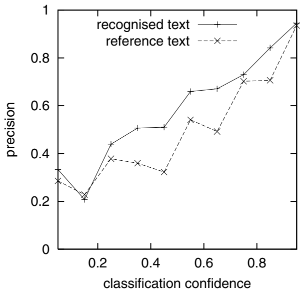

Figure 1: *Precision versus classification confidences.*

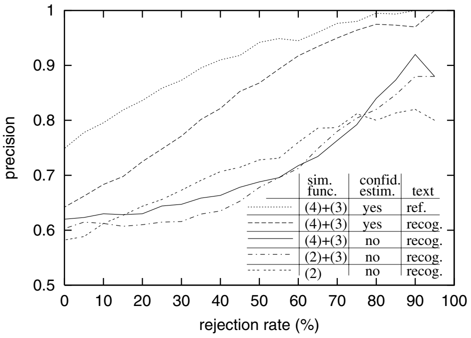

Figure 2: *Comparison of ROC curves. Lines are guides*

5.2. Classification performance

In figure 2, the performance of the combined system (speech recognition + info retrieval + classification confidence estimation) is given. Its total precision (i.e. at 0% rejection) is 0.64. Without classification confidence estimation, we obtain a slightly reduced total precision of 0.62 but the area under the ROC curve is significantly smaller. The curves with lowest total precisions 0.60 and 0.58 (last two in the table) have been obtained using the standard information retrieval setup with and without self score normalization (section 3.3), respectively.

5.3. Influence of recognition errors

To demonstrate the influence of recognition errors, we have also plotted the performance of the best classification setup on reference transcription in figure 2. While the total precision based on the reference text is with 0.75 significantly improved, the performance difference at high rejection rates is less significant.

The dependence of the total classification precision on the recognizer performance is shown in figure 3. The four data points correspond to reference transcriptions and the three recognizer setups of table 1.

A linear fit through the data points with classification confidence yields a precision slope of about $-0.003/\%$. The lower limit of precision of 0.31, which can still be reached when recognition performance breaks down completely, is given by the fraction of occurrences of the most frequent class in the test material. We would expect the precision of our classifier to converge to the same value.

We have also investigated whether the negative influence of recognition errors on retrieval performance can be decreased using more information producible by a speech recognizer than just the first best recognized text. In fact, as reported in [8] we found out that using word lattices in combination with confidence measures allows to diminish the gap between retrieval performance on perfect transcriptions and recognizer output. In [9] similar results on the OASIS call classification task are given, also applying vector space modeling based information retrieval on lattices with confidence measures.

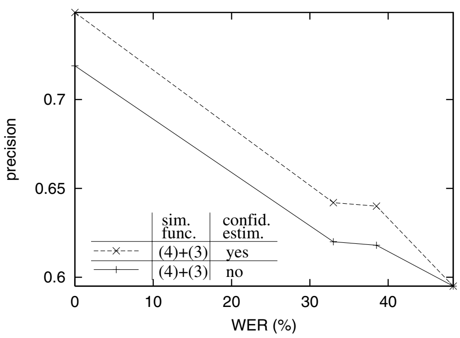

Figure 3: *Dependence of precision on WER.*

6. Conclusions

The classical vector space modeling approach to information retrieval has been extended to the framework of call classification. We have quantified the correlation between accuracy of the speech recognizer and classification accuracy as well as the contribution of various recognizer improvements made to recognition accuracy. We introduced a strategy for classification confidence estimation that turned out to produce realistic values and that improved the shape of measured ROC curves.

7. Acknowledgments

We would like to thank Mark Farrell, David Attwater, James Allen and Peter Durston from BT for not only for providing the databases but also for their continuous support and helpful discussions.

8. References

[1] Edgington, M., Attwater, D.J. and Durston, P.J., “OASIS - a framework for Spoken Language Call Steering”, in Proc. Eurospeech, 1999.

[2] Chou, W.,Attwater, D.J. *et al.*, “Natural Language Call Steering for Service Applications”, Proc. ICSLP (Beijing, China), October 2000.

[3] Durston, P.J., Farrell, M., Attwater, D., Allen, J. , Kuo, H.-K.J., Afify, M., Fosler-Lussier, E., Lee C.-H., “OASIS Natural Language Call Steering Trial”, Proc. Eurospeech, vol. 2, p.1323-1326, September 2001.

[4] van Rijsbergen, C. J., “Information Retrieval”, Butterworths, London (1979).

[5] Carpenter, B. and Chu-Carroll J., “Vector-Based Natural Language Call Routing”, Comp. Linguist., vol. 25, 1999.

[6] Cox, S. and Shahshahani, B., “A Comparison of some Different Techniques for Vector Based Call-Routing”, in Proc. Eurospeech, vol, 4, p.2337-2340, September 2001.

[7] Salton, G. and McGill, M., “Introduction to Modern Information Retrieval”, McGraw-Hill, NY, 1983.

[8] M. Harris, J. Kneissler, F. Thiele, “Information retrieval in the audio domain”, to appear in “Algorithms in Ambient Intelligence”, Philips Research Book Series, Kluwer.

[9] J. Allen, D. Attwater, P. Durston, M. Farrell, “Improving ‘How May I Help You?’ systems using the output of recognition lattices.”, BTExact Technologies, preprint, March 2003.