Multi-slot semantics for natural-language call routing systems

Johan Boye and **Mats Wirén**
TeliaSonera R&D
Vitsandsgatan 9
SE-123 86 Farsta, Sweden
johan.boye@teliasonera.com, mats.wiren@teliasonera.com

Abstract

Statistical classification techniques for natural-language call routing systems have matured to the point where it is possible to distinguish between several hundreds of semantic categories with an accuracy that is sufficient for commercial deployments. For category sets of this size, the problem of maintaining consistency among manually tagged utterances becomes limiting, as lack of consistency in the training data will degrade performance of the classifier. It is thus essential that the set of categories be structured in a way that alleviates this problem, and enables consistency to be preserved as the domain keeps changing. In this paper, we describe our experiences of using a two-level multi-slot semantics as a way of meeting this problem. Furthermore, we explore the ramifications of the approach with respect to classification, evaluation and dialogue design for call routing systems.

1 Introduction

Call routing is the task of directing callers to a service agent or a self-service that can provide the required assistance. To this end, touch-tone menus are used in many call centers, but such menus are notoriously difficult to navigate if the number of destinations is large, resulting in many misdirected calls and frustrated customers. Natural-language call routing provides an approach to come to terms with these problems. The caller gets the opportunity to express her reasons for calling using her own words, whereupon the caller's utterance is automatically categorized and routed.

This paper focuses on experiences obtained from the deployment of a call-routing application developed for the TeliaSonera residential customer care.[1] The application was launched in 2006, replacing a previous system based on touch-tone menus. The customer care annually handles some 14 million requests and questions concerning a wide range of products in fixed telephony, mobile telephony, modem-connected Internet, broadband, IP telephony and digital TV.

The crucial step in any call routing application is classification, that is, the mapping of natural-language utterances to categories that correspond to routing destinations. Early systems used quite small numbers of categories. For example, the original "How May I Help You" system had 15 categories (Gorin et al. 1997), the system of Chu-Carroll and Carpenter (1999) had 23 categories, and Cox and Shahshahani (2001) had 32. Nowadays, it is possible to distinguish between several hundreds of categories with high accuracy (see, for example, Speech Technology Magazine 2004). The TeliaSonera system currently distinguishes between 123 categories with an accuracy of 85% (using a speech recognizer and classifier developed by Nuance[2]). Moreover, according to our experiments the same classification technology can be

[1] TeliaSonera (www.teliasonera.com) is the largest telecom operator in the Nordic–Baltic region in Europe.
[2] www.nuance.com.

Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, pages 68–75,
NAACL-HLT, Rochester, NY, April 2007.

used to distinguish between 1,500 categories with 80% accuracy.[3]

For large category sets like these, the problem of maintaining consistency among manually tagged utterances becomes limiting, as lack of consistency in the training data will degrade performance of the classifier. The problem is exacerbated by the fact that call-routing domains are always in a state of flux: Self-services are being added, removed, modified, split and merged. Organizational changes and product development regularly call for redefinitions of human expertise areas. All of these changes must be accommodated in the category set. Hence, it must be possible to update this set efficiently and at short intervals.

To meet this problem, it is crucial that the set of categories be structured in a way that facilitates the task of manual tagging and enables consistency to be preserved. However, in spite of the fact that the size of category sets for call routing have increased dramatically since the original "How May I Help You" system, we are not aware of any papers that systematically discuss how such large sets should be structured in order to be efficiently maintainable. Rather, many papers in the call-routing literature consider the call routing problem as an abstract classification task with atomic categories at a single level of abstraction. Such atomic categories are typically taken to correspond to departments and self-services of the organization to which the call center belongs. In a real-life implementation, the situation is often more complicated. At TeliaSonera, we have adopted a two-level multi-slot semantics as a way of maintaining modularity and consistency of a large set of categories over time.

The aim of this paper is to share our experiences of this by providing a detailed description of the approach and its implications for classification, dialogue design and evaluation. The rest of the paper is organized as follows: Section 2 describes the multi-slot category system. Sections 3–5 outline consequences of the multi-slot semantics for disambiguation, classification and evaluation, respectively. Section 6 concludes.

2 What's in a category?

2.1 Motivation

As pointed out above, call-routing domains are always to some extent moving targets because of constant changes with respect to products and organization. It would be cumbersome to manually re-tag old data each time the category set is updated. Retagging the training data for the statistical classifier might introduce inconsistencies into the training set and degrade classifier performance. Thus, it is a good idea to define *two* sets of categories at different levels; one set of *semantic* categories reflecting the contents of the utterance, and one set of *application* categories reflecting how the call should be handled. These two sets of categories are related by means of a many-to-one mapping from the semantic domain to the application domain. Figure 1 gives the general picture.

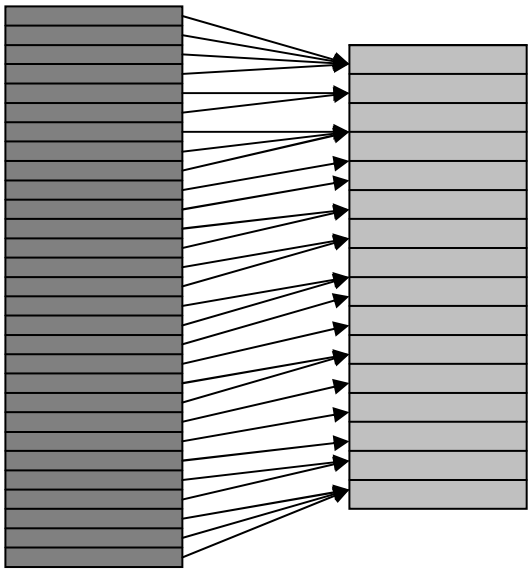

Figure 1: Mapping between semantic categories and application categories.

The utterances in the training set for the automatic classifier are manually categorized using semantic categories. The automatic classifier can be trained to work either in the semantic domain or in the application domain (see further Section 4).

[3] In both cases, the classifier was trained on 60,000 utterances.

2.2 Semantic categories

In the TeliaSonera system, semantic categories are triples of the form

(*family, intention, object*)

where *family* is the general product family which the call concerns (e.g. fixed telephony, mobile telephony, broadband, etc.), *intention* represents the nature of the request (e.g. order, want-info, change-info, activate, want-support, report-error, etc.), and *object* represents more specifically what the call is about (e.g. particular names of products, or concepts like "telephone number", "SIM card", or "password"). Currently there are 10 families, about 30 intentions, and about 170 objects that span the semantic domain.

Some (in fact, the majority) of the possible triples are disallowed because they are nonsensical. For instance, it is not meaningful to combine "fixed telephony" in the *family* slot with "SIM card" in the *object* slot. To cater for this, we have defined a set of combination rules weeding out the illegal combinations of values. These rules disallow about 80% of the possible combinations, leaving about 10,000 permissible semantic triples. Of these 10,000 triples, about 1,500 have actually turned up in real data.

The three-slot structure of categories is very useful when performing manual tagging of the training material for the statistical classifier. Although there are 10,000 categories, the person performing the tagging needs only to keep track of about 210 concepts (10 families + 30 intentions + 170 objects). In contrast, it is safe to say that an unstructured category system containing 10,000 atomic categories would be quite impractical to use.

In addition, the combination rules can further alleviate the manual tagging task. It is straightforward to implement a tagging tool that allows the human tagger to select a value for one semantic slot, and then restrict the selection for the other slots only to include the possible values. For example, if "fixed telephony" is chosen for the *family* slot, "SIM card" would not appear among the possible values for the *object* slot. This approach has been successfully adopted in the project.

2.3 Application categories

There is one application category for each type of action from the system. Actions come in two flavors; either the call is routed (in the cases where the caller has given sufficient information), or the system asks a counter-question in order to extract more information from the caller. That is, application categories can be labeled either as *routing categories* or *disambiguation categories*. For convenience, names of application categories are also triples, chosen among the set of semantic triples that map to that application category.

2.4 Information ordering

Each slot in a semantic triple can take the value *unknown*, representing the absence of information. For instance, the most accurate semantic category for the caller utterance "Broadband"[4] is (*broadband*, *unknown*, *unknown*), since nothing is known about the intention of the caller or the specific topic of the request. Thus, in the information ordering, "unknown" is situated below all other values.

There are also some intermediate values in the information ordering. The value *telephony* represents "either fixed telephony or mobile telephony", and has been incorporated in the category set since many callers tend not be explicit about this point. In the same vein, *internet* represents "either broadband or modem-connected internet", and *billing* represents the disjunction of a whole range of billing objects, some of which can be handled by a self-service and some can not.

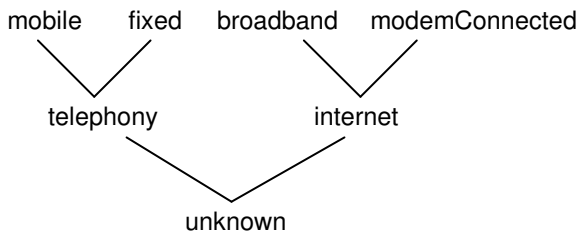

Figure 2: Parts of the semantic information ordering.

The information ordering extends naturally to triples. In particular, the triple (*unknown*, *unknown*,

[4] Many callers express themselves in this telegraphic fashion.

unknown) represents complete absence of information.

3 Disambiguation

The caller's request might be ambiguous in one sense or another, in which case the system will need to perform disambiguation by asking a follow-up question. This might either be a general question encouraging the user to describe his request in greater detail, or a directed question of the type "Would that be fixed telephony or mobile telephony?"

Ambiguous utterances might be represented in at least two fundamentally different ways. In vector-based approaches, routing destinations and input utterances alike are represented by vectors in a multi-dimensional space. An input utterance is routed to a specific destination if the vector representation of the utterance is close to that of the destination. An ambiguous utterance is characterized by the fact that the Euclidean distances from the utterance vector to the *n* closest routing destination vectors are roughly the same.

Chu-Carroll and Carpenter (1999) describe a method of disambiguation, where disambiguation questions are dynamically constructed on the basis of an analysis of the differences among the closest routing destination vectors. However, it is not clear that the disambiguation questions produced by their proposed method would make sense in all possible situations. Furthermore, their method does not take into account the fact that some ambiguities tend to be more important and arise more often than others. We think it is worthwhile to concentrate on these important cases (in terms of prompt design, speech recognition grammar construction, etc.), rather than trying to solve every conceivable ambiguity, most of which would never appear in real life.

As previously mentioned, in the TeliaSonera system we have chosen another way of treating ambiguities, namely that certain application categories are *disambiguation categories*; they represent foreseen, frequently occurring, ambiguous input utterances. The three-slot structure of categories provides a handy way of identifying ambiguous cases; they are represented by triples where one or more slots are *unknown,* or where some slot has an intermediate value, like *telephony* or *internet.* Examples of such ambiguous utterances are "broadband" (*broadband-unknown-unknown*) and "I want to have a telephone subscription" (*telephony-order-subscription*). All categories that represent ambiguities have pre-prepared disambiguation questions, speech recognition grammars, and dialogue logic to handle the replies from the callers.

Of course, there are still problematic cases where an utterance can not be assigned any unique category with any tolerable level of confidence, neither a routing category nor a disambiguation category. In those cases, the system simply rephrases the question: "*Sorry, I didn't quite understand that. Could you please rephrase?*"

4 Classification

4.1 Atomic vs. multi-slot classification

For the purpose of automatic classification of utterances, there are at least two different views one may adopt. In one view, the "atomic" view, the three-slot structure of category names is considered as merely a linguistic convention, convenient only when manually tagging utterances (as discussed in Section 2.1). When adopting this view, we still regard the categories to be distinct atomic entities as concerns automatic classification. For instance, to the human eye it is obvious that two categories like (*internet, order, subscription*) and (*broadband, order, subscription*) are related, but the automatic classifier just considers them to be any two categories, each with its separate set of training examples.

An alternative view, the "multi-slot view", is to see the category as actually consisting of three slots, each of which should be assigned a value independently. This means that a separate classifier is needed for each of the three slots.

It is not clear which view is preferable. An argument in favor of the multi-slot view is the following: If some categories have the same value in one slot, then these categories are semantically related in some way. Most likely this semantic relation is reflected by the use of common words and phrases; for instance, expressions like "order" and "get a new" presumably are indicative for all categories having the value *order* in the *intention* slot. Therefore, classifying each slot separately would be a way to take a priori semantic knowledge into account.

To this, proponents of the atomic view may respond that such similarities between categories

would emerge anyway when using a single classifier that decides the entire semantic triple in one go (provided that enough training data is available). In addition, if each slot is categorized separately, it is not certain that the resulting three values would constitute a permissible semantic triple (as mentioned in Section 2.1, about 80% of the possible combinations are illegal). In contrast, if a single classifier is used, the result will always be a legal triple, since only legal triples appear in the training material.

The statistical classifier actually used in the live call routing system treats categories as atomic entities and, as mentioned in the introduction, it works well. The encouraging numbers bear out that the "atomic" view is viable when lots of data is at hand. On the other hand, if training data is sparse, one might consider using a hand-written, rule-based classifier, and in these cases the multi-slot view seems more natural.

4.2 Rule-based multi-slot classification

To obtain a baseline for the performance of the statistical classifier used in the live system, we implemented an alternative classifier that solves the classification task using hand-written rules. Thus, the purpose of this was to investigate the performance of a naïve classification method, and use that for comparison with other methods. In addition, the rule-based classifier provides an example of how the multi-slot approach can support the inclusion of human a priori domain knowledge into the classification process.

The rule-based classifier has three kinds of rules: Firstly, phrase-spotting rules associate a word or a phrase with a value for a semantic slot (i.e. a *family*, an *intention*, or an *object*). Rules of the second kind are domain axioms that encode invariant relationships, such as the fact that *object=SIMcard* implies *family=mobileTelephony*. Finally, rules of the third kind specify how semantic values can be combined into a legal semantic triple (these rules are also used for manual tagging, as mentioned in Section 2.1). Each semantic value is also (manually) given a score that reflects its information content; a higher score means that the value contains more information. For instance, the value *subscription* has a lower information score than have the names of specific subscription types that TeliaSonera offers its customers.

The classifier works in three phases, which we will demonstrate on a running example. In the first phase, it applies the phrase-spotting rules to the input sentence, returning a list of slot-value pairs. For instance, the input sentence "I want to order a new SIM card" would yield the list [*intention=order, object=SIMcard*], using rules triggering on the phrases "order" and "SIM card" in the input sentence.

Secondly, the classifier adds semantic components as a result of applying the domain axioms to members of the list. Using the domain axiom mentioned above, the semantic component *family=mobileTelephony* would be added to the list, due to the presence of *object=SIMcard.* Thus, after the two first phases, the intermediate result in this example is [*intention=order, object=SIMcard, family=mobileTelephony*].

In the final phase, semantic components are selected from the list to form a semantic triple. In the example, this step is straightforward since the list contains exactly one value for each component, and these values are combinable according to the combination rules. The final result is:

(*mobileTelephony, order, SIMcard*)

In cases where the semantic values in the list are not combinable (a situation often originating from a speech recognition error), one or several values have got to be relaxed to *unknown.* According to our experiments, the best heuristic is to first relax the *object* component and then the *intention* component. For example, in the list [*family = fixedTelephony, intention=order, object=SIMcard*], the first and third elements are not combinable; thus this list yields the triple:

(*fixedTelephony, order, unknown*)

In the case where some slots are not filled in with a value, the values of those slots are set to *unknown.* Thus, the list [*family=fixedTelephony, intention=order*] would also yield the semantic triple above.

Finally, consider the case where the input list contains more than one value for one or several slots. In this case, the algorithm picks the value with the highest information content score. For instance, consider the utterance "I want to have a broadband subscription, this eh ADSL I've read

about". After the first two phases, the algorithm has found *family=broadband, intention=order,* and two possible values for the *object* slot, namely *object=subscription* and *object=ADSL.* Since the latter has higher information score, the final result is:

(*broadband, order, ADSL*)

The rule-based classifier was developed in about five man-weeks, and contains some 3,000 hand-written rules. When evaluated on a set of 2,300 utterances, it classified 67% of the utterances correctly. Thus, not surprisingly, its performance is significantly below the statistical classifier used in the deployed system. Still, the rule-based approach might be a viable alternative in less complex domains. It might also be usable for data collection purposes in early prototypes of natural-language call routing systems.

5 Evaluation of call-routing dialogues

5.1 Motivation

An important issue in the development of any dialogue system is the selection of an evaluation metric to quantify performance improvements. In the call-routing area, there have been many technical papers specifically comparing the performance of classifiers, using standard metrics such as accuracy of the semantic categories obtained over a test corpus (see e.g. Kuo and Lee, 2000, and Sarikaya et al., 2005). Accuracy is then stated as a percentage figure showing the degree of the categories that have been completely correctly classified, given that categories are atomic. There have also been some design-oriented papers that try to assess the effects of different prompt styles by looking at the proportion of routable versus unroutable calls given callers' first utterances. Thus, both of these strands of work base their evaluations on binary divisions between correct/incorrect and routable/unroutable, respectively. Furthermore, they both constitute utterance-based metrics in the sense that they focus on the outcome of a single system–caller turn.

An excellent example of a design-oriented call-routing paper is Williams and Witt (2004), which among other things compares open and directed prompt styles in the initial turn of the dialogue. Williams and Witt divide callers' responses into *Routable* (if the utterance contained sufficient information for the call to be routed) or *Failure* (if the utterance did not contain sufficient information for routing). Depending on why a call is not routable, Williams and Witt further subdivide instances of *Failure* into three cases: *Confusion* (utterances such as "Hello?" and "Is this a real person?"), *Agent* (the caller requests to speak to a human agent), and *Unroutable* (which corresponds to utterances that need disambiguation). Thus, Williams and Witt's performance metric uses altogether four labels. (In addition, they have three labels related to non-speech events: silence, DTMF and hang-up. Since such events are not handled by the classifier, they fall outside of the scope of this paper.)

Although all of Williams' and Witt's measures are needed in evaluating call-routing dialogue, the field clearly needs more in-depth evaluation. In particular, we need *more fine-grained metrics* in order to probe more exactly to what extent *Failure* actually means that the dialogue is off track. Furthermore, given that call-routing dialogues typically consist of between one and (say) five turns, we need not just utterance-based metrics, but also *dialogue-based metrics* — in other words, being able to evaluate the efficiency of an overall dialogue.

5.2 Utterance-based metrics

When assessing the performance of classification methods, it is perfectly reasonable to use the binary distinction correct/incorrect if only few categories are used. In such a context it can be assumed that different categories correspond to different departments of the organization, and that a misclassification would lead the call being routed the wrong way. However, with a richer category system, it is important to realize that the classifier can be partially correct. For instance, if the caller expresses that he wants technical support for his broadband connection, then the information that the purpose of the call has something to do with broadband is surely better than no information at all. If the system obtains this information, it could ask a directed follow-up question: *OK broadband. Please tell me if your call concerns an order, billing, deliveries, support, error report, or something else,* or something to that effect. Otherwise, the system can only restate the original question.

In the field of task-oriented dialogue, several evaluation metrics have been put forward that go beyond a simple division into correct/incorrect. In particular, *concept accuracy* (Boros et al. 1996) is an attempt to find a semantic analogue of word accuracy as used in speech recognition. Basically, the idea is to compute the degree of correctness of a semantic analysis based on a division of the representation into subunits, and by taking into account insertions, deletions and replacements of these subunits.

Making use of our multi-slot semantics, we can take subunits to correspond to semantic slot values. An insertion has occurred if the classifier spuriously has added information to some slot value (e.g. if the classifier outputs the value *broadband* for the *family* slot, when the correct value is *internet* or *unknown*). Conversely, a deletion has occurred when semantic triple output from the classifier contains a slot value which is situated lower than the correct value in the information ordering (a part of which is depicted in Figure 2). Finally, a replacement has occurred when the computed slot value and the correct slot value are unrelated in the information ordering.

By using concept accuracy as an evaluation metric for classifiers rather than the binary distinction correct/incorrect, we can arrive at more informative assessments. This possibility is brought about by the multi-slot structure of categories.

5.3 Dialogue-based metrics

In the literature, there have also been proposals for dialogue-based metrics. In particular, Glass et al. (2000) put forward two such metrics, *query density* (*QD*) and *concept efficiency* (*CE*). Query density is the mean number of new "concepts" introduced per user query, assuming that each concept corresponds to a slot–filler pair in the representation of the query. For example, a request such as "I'd like a flight from Stockholm to Madrid on Sunday afternoon" would introduce three new concepts, corresponding to departure, destination and time. Query density thus measures the rate at which the *user communicates content*. In contrast, concept efficiency measures the average number of turns it takes for a concept to be successfully understood by the system. Concept efficiency thus measures the rate at which the *system understands content*.

Using the multi-slot semantics, we can adapt the notions of query density and concept efficiency in order to arrive at a more fine-grained performance metric for call routing. The basic idea is to regard every element in the semantic triple as one "concept". We can then obtain a measure of how information increases in the dialogue by computing the difference between triples in each user utterance, where "difference" means that the values of two corresponding elements are not equal.

An example of computing query density is given below. We assume that the value of the semantic triple is initially (*unknown*, *unknown*, *unknown*).

System: Welcome to TeliaSonera. How may I help you?
Caller: Fixed telephony.
(fixedTelephony, unknown, unknown)
`1 new concept`
System: Could you tell me some more about what you want to do?
Caller: I can't use my broadband while I'm speaking on the phone.(*broadband, reportProblem, lineOrPhone*)
`3 new concepts`

Note that query density and concept efficiency are both applicable on a per-utterance basis as well as on the whole dialogue (or indeed arbitrary stretches of the dialogue). To compute these measures for the whole dialogue, we simply compute the mean number of new concepts introduced per user utterance and the average number of turns it takes for a concept to be successfully understood, respectively.

The principal application of this methodology is to measure the effectiveness of system utterances. When using a fine-grained system of categories, it is important that callers express themselves at a suitable level of detail. Too verbose user utterances are usually difficult to analyse, but too telegraphic user utterances are not good either, as they most often do not contain enough information to route the call directly. Therefore it is very important to design system utterances so as to make users give suitably expressive descriptions of their reasons for calling.

By using the query density metric it is possible to asses the effectiveness (in the above sense) of different alternative system utterances at various points in the dialogue, most notably the first sys-

tem utterance. Again, this possibility is brought about by the multi-slot structure of categories. It is also possible to evaluate more general dialogue strategies over longer stretches of dialogue (e.g. the use of general follow-up questions like "*Could you please tell me some more about what you want to do*" as opposed to more directed questions like "*Please tell me if your call concerns an order, billing, deliveries, support, error report, or something else*"). By calculating the average query density over a number of consecutive utterances, it is possible to compare the relative merits of different such dialogue strategies.

We have not yet adopted this metric for evaluation of dialogues from the live system. However, elsewhere we have applied it to dialogues from the initial Wizard-of-Oz data collection for the TeliaSonera call routing system (Wirén et al. 2007). Here, we used it to compare two styles of disambiguation prompts, one completely open and one more directed.

6 Concluding remarks

In the literature, the natural-language call routing problem is often presented as the problem of classifying spoken utterances according to a set of atomic categories. The hypothesis underlying this paper is that this view is inadequate, and that there is a need for a more structured semantics. We base our claims on experiences gathered from the development and deployment of the TeliaSonera call center, for which we developed a multi-slot system of categories.

A multi-slot semantics offers several advantages. First of all, it makes the set of categories manageable for human taggers, and provides a means to break down the tagging task into subtasks. Furthermore, we have shown how multi-slot semantics for call-routing systems allows straightforward division of categories into routing categories and disambiguation categories, the possibility of multi-slot categorization, and the use of more fine-grained evaluation metrics like concept accuracy and query density.

Acknowledgements

This work has benefited greatly from discussions on category systems and classification with Marco Petroni, Linda Broström, Per-Olof Gällstedt, Alf Bergstrand and Erik Demmelmaier, and we thank them all. We would also like to thank Robert Sandberg and Erik Näslund for their support of this work.

References

Boros, M., Eckert, W., Gallwitz, F., Görz, G., Hanrieder, G. and Niemann, H. (1996). Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. *Proc. Fourth International Conference on Spoken Language Processing* (*ICSLP*), pp. 1009–1012.

Chu-Carroll, J. and Carpenter, B. (1999) Vector-based natural language call routing. *Computational linguistics,* 25(3), pp. 361-388.

Cox, S. and Shahshahani, B. (2001). A comparison of some different techniques for vector based call-routing. *Proc. Eurospeech*, Aalborg, Denmark.

Glass, J., Polifroni, J., Seneff, S. and Zue, V. Data collection and performance evaluation of spoken dialogue systems: The MIT experience. In *Proc. Sixth International Conference on Spoken Language Processing* (*ICSLP*), Beijing, China.

Gorin, A., Riccardi, G., and Wright, J. (1997) How may I help you?. *Journal of Speech Communication,* 23, pp. 113-127.

Kuo, H-K J. and Lee, C-H. (2000) Discriminative training in natural language call routing. *Proc. Sixth International Conference on Spoken Language Processing* (*ICSLP*), Beijing, China.

Sarikaya, R, Kuo, H-K J., Goel, V. and Gao, Y. (2005) Exploiting unlabeled data using multiple classifiers for improved natural language call-routing. *Proc. Interspeech*, Lisbon, Portugal.

Speech Technology Magazine (2004) Q&A with Bell Canada's Belinda Banks, senior associate director, customer care. *Speech Technology Magazine,* vol 9, no 3.

Williams, Jason D. and Witt, Silke M. (2004). A comparison of dialog strategies for call routing. *International Journal of Speech Technology* 7(1), pp. 9–24.

Wirén, M., Eklund, R., Engberg, F. and Westermark, J. (2007). Experiences of an in-service Wizard-of-Oz data collection for the deployment of a call-routing application. *Proc. Bridging the gap: Academic and industrial research in dialog technology. NAACL workshop*, Rochester, New York, USA.