Hybrid Vector Space Model for Flexible Voice Search

Cheongjae Lee and Tatsuya Kawahara
Academic Center for Computing and Media Studies, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan

***Abstract*—This paper addresses incorporation of semantic analysis into information retrieval (IR) based on the vector space model (VSM) for flexible matching of spontaneous queries in a voice search system. Information of semantic slots or concepts that correspond to database fields is expected to help enhancing IR, but the semantic analyzer often fails or needs a large amount of training data. We propose a hybrid model which combines dedicated VSMs for concept slots with a general VSM as a back-off. The model has been evaluated in a book search task and shown to be effective and robust against ASR and SLU errors.**

Index Terms: spoken language understanding, voice search, vector space model

I. Introduction

In the past years, voice search has become one of the most popular applications of the spoken language technology [1]. It has caused a paradigm shift in spoken language understanding (SLU) from semantic slot filling to information retrieval (IR) [2][3] as the data to be matched are not structured and very large in size. IR can be generally modeled with document-based vector space models (VSM) or language models (LM) [4].

However, there are a number of applications in which IR should be used, but still we can define semantic slots or concepts that correspond to fields of a large database. In addition to simple directory services [5], typical examples include music search [6] and book search [7]. In these applications, titles, authors or artists, and categories are referred by users in a single query of a variety of patterns. Moreover, users often do not exactly remember long titles and make references in inaccurate ways.

These pose a necessity of a novel SLU scheme which combines semantic analysis and IR techniques. Feng et al. [5] investigated query parsing methods based on a statistical scheme for directory search. Song et al. [6] proposed the HMMIR model for music search, in which a semantic concept is regarded as a hidden state, which makes transition and generates query words. Lee et al. [7] proposed a tandem approach for book search, which first applies a rule-based parser and then feeds its results separately to respective VSMs. However, the semantic analysis often fails because of the lack of coverage or training data needed for a large variety of the queries. In these cases, the entire search will not appropriately work.

In this paper, we present a hybrid approach which uses the concept slot-dependent VSMs and the standard single VSM in parallel, and also making the latter as a back-off. Moreover, we introduce a semantic tagger based on Conditional Random Fields (CRF) [8]. These are intended to realize robust SLU and flexible matching. The method is evaluated in book search queries collected via Amazon's Mechanical Turk (MTurk).

II. Voice Book Search Task

The book search task [7] poses interesting challenges for spoken language processing, as the core interaction involves search for an often under-specified item, a book for which the user may have incomplete or inaccurate information. For example, a typical query might be "I AM LOOKING FOR ALICE IN WONDER-LAND BY CARROLL", while the exact title of the book is "ALICE'S ADVENTURES IN WONDER-LAND AND THROUGH THE LOOKING-GLASS". In our preliminary survey conducted via MTurk, 33% of 200 respondents did not have the complete information. In fact, many titles are simply too long to say even though users might know exactly (in our database the longest title has 38 words). Thus, users often provide a few keywords instead of the exact title. There are additional peculiarities, for instance, the book entitled "MISS PARLOA'S NEW COOK BOOK" contains its author's name and the category in its title. This causes ambiguity and degradation in extracting semantic concepts, exacerbated by the large number of book entries.

Our system uses a relational database (RDB) consisting of 15,088 eBooks, sampled randomly from the Amazon Kindle Book website. Each book entry can have up to 17 attributes including its title, authors, category, price, and sales rank. Based on our preliminary survey on which information is used for book search, we selected the top-three attributes of title, author and category as concept slots in this work. These kinds of semantic information should be used in analyzing queries and matching against database entries. Since there are a variety of patterns in queries, robust SLU and IR models are vitally important.

III. Robust IR Model

We first review the conventional VSM, and then the multiple VSM which incorporates semantic analysis, followed by the hybrid model, which is proposed in this paper.

A. Baseline Vector Space Model (VSM)

A VSM assumes a term space, in which each database entry (=book) i is represented as a vector of term occurrences with

specific weights in a high-dimensional space (v_i). A query q is also represented as the same kind of vector (v_q). A retrieved list of books is created by calculating the cosine similarity $s(v_q, v_i)$ between the two vectors as below:

$$s(v_q, v_i) = \frac{v_q \cdot v_i}{\|v_q\|\|v_i\|} \quad (1)$$

When the vectors are normalized, it is possible to compute the cosine similarity as the dot product between the unit vectors.

$$s(v_q, v_i) = \hat{v}_q \cdot \hat{v}_i \quad (2)$$

This formulation allows for rapid search, which is important since there are many vectors to compute for each query.

We use stemming to compact the representation, but we do not eliminate stop words as some stop words are necessary and meaningful for identifying relevant books. For example, some titles consist of only stop words such as "YOU ARE THAT" and "IT". They will not be indexed correctly if stop words are filtered out.

There are several different ways of assigning term weights, but not all are appropriate for this task. For example, TF-IDF does not work well for book search since most of entries and queries are too short to estimate reliable weights. Thus, we used a simple term count for a weight.

In the conventional single VSM (hereafter, S-VSM), all terms in different concept slots are indexed together over a single term space and all terms are equally weighted regardless of their semantic attributes. This model can be robust against SLU errors in which concepts are incorrectly extracted. However, it cannot capture relationships between concepts from queries such as "A MYSTERY BY CHRISTIE".

B. Concept slot-based Multiple VSM

We also consider a multiple VSM model (M-VSM) in which semantic analysis is conducted to extract concepts and corresponding tagged sequences of words. In this work, we consider three slots of title, authors and category. The concept slot-based query v_{qj} is generated by each slot value, for example, "ALICE ADVENTURE" for the title query and "CARROLL" for the author query. Each concept slot c has a dedicated VSM to compute the cosine similarity $s_c(v_{qc}, v_{ic})$ for a query q and a database entry i. The final decision of IR is made based on the weighted sum of the slot-based models.

$$s(v_q, v_i) = \sum_c w_c \cdot s_c(v_{qc}, v_{ic}) \quad (3)$$

In this work, the slot weights w_c are set based on the slot preferences that users expressed in our previous survey [7], so a more frequently-used slot has a larger value. It can be tuned to improve the IR performance using held-out data, or dynamically modified based on the user's preference or confidence scores of ASR and SLU, but these issues are for future work. Weights for un-filled slots are set to 0, and other valid weights are normalized dynamically so that their sum makes 1.

A classical way to extract concepts is to use a parser based on hand-crafted grammars. We initially adopted the PHOENEX parser [9], which was shown to be robust in the conventional spoken dialogue applications including ATIS. However, the rule-based parser often fails in this domain. Therefore, we introduce a CRF-based tagger [8][10]. CRF is trained to tag a word sequence with the particular concept tag, one of the three slots in this work. The statistical model is expected to work more robustly, though it needs a large amount of training data.

If the semantic analysis works properly, the M-VSM will perform better. On the other hand, the ambiguity or errors in SLU may degrade its performance.

C. Hybrid VSM and Back-off Scheme

Finally, we propose a hybrid VSM (H-VSM) in which the S-VSM and the M-VSM are used in parallel. When some concept slots are filled by the parser or tagger and the M-VSM is applied effectively, the final decision is made based on the sum of evaluation scores by the two models. Here, a weighted interpolation could be explored using held-out data, but is not tried in this work. If no concept slots are obtained due to the lack of coverage or errors in input, only the S-VSM is applied. Namely, the S-VSM is used as a back-off when the semantic analysis fails.

The whole search strategy is illustrated in Figure 1. First, a semantic analyzer extracts domain-specific concepts from the ASR hypothesis for an input spoken query. Once any slots are identified, two different types of queries are generated: slot-independent query (e.g. v_q=[ALICE, ADVENTURE, CARROLL]) and slot-dependent queries (e.g. $v_{q,TITLE}$=[ALICE ADVENTURE], $v_{q,AUTHOR}$=[CARROLL]). These queries are matched with the S-VSM and the M-VSM, respectively. Finally, the returned items are merged into a single list based on the sum of individual matching scores. If no concept slot is filled, all terms are used to generate a single vector as a query to the S-VSM.

We expect that this hybrid model can compensate for the individual drawbacks of the S-VSM and the M-VSM at the cost of additional computation.

IV. Speech Data Collection via MTurk

We turned to Amazon's Mechanical Turk (MTurk) for collection of spoken queries as well as textual queries for experimental evaluations, because it is not easy to collect many native English speakers in the authors' countries (Japan and Korea).

We created a HIT (Human Intelligence Task) that elicits utterances by providing metadata consisting of a book title, authors, and a category. Turkers were asked to formulate a response to the question "how can I help you?" posed by a hypothetical bookstore clerk. First, we collected 1574 typed or textual queries.

Collection of speech data is more challenging [11] because the audio recording conditions of the subjects cannot be controlled and data need to be transcribed. We used the web-based recording interface in Java applet provided by VIMAS [1].

[1] http://www.vimas.com/

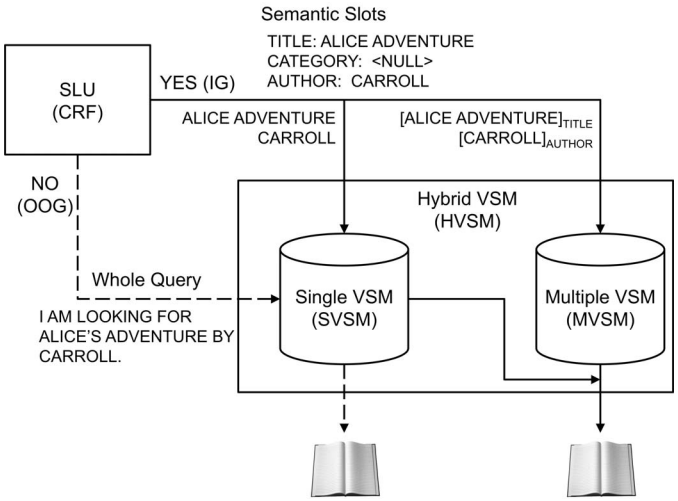

Fig. 1. Overview of hybrid IR model (IG: In-grammar, OOG: Out-of-grammar)

This enables turkers to record audio data, which are sent to the web server. After the utterance, turkers were asked to transcribe what they had said just now. We have collected 1011 utterances from 38 different turkers in this manner.

The collected data were proofed via MTurk, too. We published HITs for evaluating the recording quality and the transcription quality. Turkers, who are different from those engaged in recording, listened to the recording of each utterance, and filled forms on the quality of the recorded audio and the transcript. Each HIT was assigned to three different turkers. The judgment was validated if two of the three turkers agreed (majority vote); otherwise, the data was marked as invalid.

As a result, 8.8% of the recorded audio files were identified as low quality, and 6.0% of the transcripts were not correct, thus they were corrected manually. However, more than 90% of the collected data are usable as they are. This might be a better scenario than real-world applications. We also evaluated non-expert turker judgments by measuring their inter-annotator agreement with the expert judgment. This yields a κ value of 0.92 in the recording quality and 0.94 in the transcription quality, which can be considered as a good agreement.

V. Experimental Evaluations

Experimental evaluations were conducted using the collected data. For search evaluation, we adopt two evaluation metrics widely used in IR. One is precision at n ($P@n$), which represents the number of queries having the correct answer in the top n retrieved items divided by the total number of queries. The other is mean reciprocal rank (MRR), which indicates the average of the reciprocal ranks of the correct search results over the entire set of queries. In reality there may be multiple correct answers in a list, when users do not have a specific book in their mind. In an extreme case, some user can make a query for any fictions, like "I AM LOOKING FOR A BOOK IN FICTION CATEGORY". We tried to annotate a single "correct" book for each query with our best effort.

TABLE I
Search evaluation on textual queries using parser

	S-VSM		M-VSM		H-VSM	
	P@100	MRR	P@100	MRR	P@100	MRR
IG	0.885	0.605	0.833	0.691	0.934	0.771
OOG	0.809	0.539	NA	NA	0.809	0.539
total	0.853	0.577	0.823	0.630	0.882	0.676

A. Rule-based Parser tested in Textual Queries

We made a preliminary evaluation to see the effect of the proposed IR model using the PHOENIX parser and textual queries. We manually created a grammar for the PHOENIX parser, but sub-grammars for the concept slot values are automatically generated from the book database. Some heuristic rules are applied, for example, to extract only keywords from the book title and last name from the author name.

Among textual queries being tested, 41.4% were not parsed due to the lack of coverage, thus regarded as OOG. For the parsed results, the concept errors rate (CER) was 24.8%. This shows that it is not easy to hand-craft a grammar to cover spontaneous queries and thus is necessary to have a back-off scheme when using the parser.

Table I lists the book search performance, with a breakdown for parsed (IG) queries and unparsed (OOG) queries. It is observed that the M-VSM which incorporates the semantic parser has better MRR but lower precision than the standard S-VSM, because more precise retrieval can be made when the parsing is successful. However, the M-VSM cannot get meaningful results for unparsed queries. The hybrid model (H-VSM) maintains the advantage of the M-VSM and is complemented by the S-VSM as a back-up. Therefore, it achieves higher precision and MRR than these two component models.

B. CRF-based Tagger tested in Spoken Queries

Next, we applied the methods to spoken queries. All 1011 collected spoken queries were used for this evaluation. At this time, we could use the textual queries, which had been collected earlier and used for the preliminary evaluation in the previous sub-section, as a training data set for the LM for ASR and the CRF-based tagger for SLU.

However, the set of 1574 queries is not sufficient to provide a good coverage for their training. Therefore, we built a trigram LM by generating sampled queries. All typed queries were manually annotated with concept slots (i.e. title, author, and category), and then 10K synthetic sentences were automatically generated from the annotated queries and the metadata of book entries. Original slot values in the queries were replaced via the metadata with those randomly selected in the back-end book database containing 15K book entries. The CRF-based semantic analyzer or concept tagger was trained in the same manner based on the annotation of the concept slots.

For the test set of spoken queries, the out-of-vocabulary rate was 1.7% and word perplexity was 92.1. The word error

TABLE II
SEARCH EVALUATION ON SPOKEN QUERIES USING CRF TAGGER (MANUAL TRANSCRIPTS AND ASR OUTPUTS)

	S-VSM		M-VSM		H-VSM	
	P@100	MRR	P@100	MRR	P@100	MRR
manual	0.977	0.860	0.981	0.864	0.983	0.882
ASR	0.782	0.595	0.763	0.588	0.798	0.615
	P@10	MRR	P@10	MRR	P@10	MRR
text	0.950	0.856	0.944	0.837	0.962	0.876
ASR	0.711	0.594	0.686	0.570	0.730	0.616

rate (WER) of the ASR results was 35.1%. The relatively high WER can be attributed to the facts that users were often in an uncontrolled environment, and they were naive users to an ASR system. But this WER would provide a realistic condition for evaluating SLU and IR in real-world voice search applications.

The accuracy of the CRF-based semantic analysis is also measured by the CER. Here the number of correct slots is calculated by using the cosine similarity between the reference and the hypothesis to allow for flexible partial matching. The CER was 10.2% for manual transcripts of the test queries and 39.9% for the ASR results. The statistical model apparently outperforms the rule-based parser, as the ratio of failure to output anything was only 5.9% and the CER was significantly reduced. But note that it needs a large amount of training data and cannot be accommodated in the proto-type.

Then, the search performance is evaluated in Table II. The table also shows the results for the case retrieving top-10 items only, which is more practical in the voice search usually done in mobile environments.

Because of the improved semantic analysis, the overall performance is much higher than the case of Table I. The concept slot-based M-VSM performs better than the standard S-VSM for the manual transcripts, but it does not for the ASR results. This result suggests that the purely SLU-based method is not robust against ASR errors. However, the hybrid model (H-VSM) gives the best performance for all cases, showing the robustness against ASR and SLU errors. The degradation from the manual transcripts to the ASR results is around $20-25\%$ in spite of the WER of 35% and the CER of 40%.

VI. CONCLUSIONS

We have investigated better IR models for realizing flexible matching for spontaneous queries in voice search applications, by incorporating semantic analysis. It is demonstrated that the semantic analysis can improve the matching performance, but it often fails, resulting in search degradation. Therefore, we have proposed a hybrid model which uses the standard VSM as a back-up. This model works stably, achieving the best performance over all experimental conditions. It is especially effective when the semantic analyzer is not prepared well due to the lack of the training data.

Future work includes improvement of robustness to ASR and SLU errors by incorporating n-best results and confidence scores into the search algorithm as well as optimizing the interpolation weights.

REFERENCES

[1] Y.-Y.Wang, D.Yu, Y.-C.Ju, and A.Acero. An introduction to voice search. *Signal Processing Magazine*, 25(3):29–38, 2008.

[2] T.Kawahara. New perspectives on spoken language understanding: Does machine need to fully understand speech? In *Proc. IEEE Workshop Automatic Speech Recognition & Understanding (ASRU)*, pages 46–50 (invited paper), 2009.

[3] G.Tur and R.De Mori. *Spoken Language Understanding*. Wiley, Chichester, 2011.

[4] M.Seltzer, Y-C.Ju, I.Tashev, Y-Y.Wang, and D.Yu. In-car media search. *Signal Processing Magazine*, 28(4):50–60, 2011.

[5] J.Feng and S.Bangalore. Query parsing for voice-enabled mobile local search. In *Proc. IEEE-ICASSP*, pages 4777–4780, 2009.

[6] Y-I.Song, Y-Y.Wang, Y-C.Ju, M.Seltzer, I.Tashev, and A.Acero. Voice search of structured media data. In *Proc. IEEE-ICASSP*, pages 3941–3944, 2009.

[7] C.Lee, A.Rudnicky, and G.Lee. Let's buy books: Finding ebooks using voice search. In *Proc. IEEE-SLT*, pages 442–447, 2010.

[8] C.Raymond and G.Riccardi. Generative and discriminative algorithms for spoken language understanding. In *Proc. INTERSPEECH*, pages 1605–1608, 2007.

[9] W.Ward. Understanding spontaneous speech: the PHOENIX system. In *Proc. IEEE-ICASSP*, pages 365–367, 1991.

[10] M. Jeong and G. G. Lee. Tri-angular chain conditional random fields. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1287–1302, 2008.

[11] M. Marge, S. Banergee, and A. I. Rudnicky. Using the amazon mechanical turk for transcription of spoken language. In *Proc. ICASSP*, pages 5270–5273, 2010.