Language Modeling for Spoken Dialogue System based on Filtering using Predicate-Argument Structures

Koichiro Yoshino[1] *Shinsuke Mori*[1] *Tatsuya Kawahara*[1]
(1) School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto, Japan.

ABSTRACT
We present a novel scheme of language modeling for a spoken dialogue system by effectively filtering query sentences collected via a Web site of wisdom of crowds. Our goal is a speech-based information navigation system by retrieving from backend documents such as Web news. Then, we expect that users make queries that are relevant to the backend documents. The relevance measure can be defined with cross-entropy or perplexity by the language model generated from the documents in a conventional manner. In this article, we propose a novel criteria that considers semantic-level information. It is based on predicate-argument (P-A) pairs and their relevance to the documents (or topic) is defined by a naive Bayes score. Experimental evaluations demonstrate that the proposed relevance measure effectively selects relevant sentences used for a language model, resulting in significant reduction of the word error rate of speech recognition as well as the semantic-level error rate.

KEYWORDS: Language Modeling, Predicate Argument Structure, Spoken Dialogue System.

Proceedings of COLING 2012: Technical Papers, pages 2993–3002,
COLING 2012, Mumbai, December 2012.

1 Introduction

The tasks of spoken dialogue systems have been extended from simple transactions to general information navigation based upon user requests. Ideally, these systems should handle not only simple, keyword-based queries that current voice search systems respond to but also vague and complex user requests related to, for example, tourist guides or news briefings. This type of application can be achieved through document retrieval in a corresponding domain. For example, we can turn to tourist guidebooks or relevant Wikipedia entries for information on the tourist domain (Misu and Kawahara, 2010). An intelligent dialogue system can be created by restricting the domain and using the knowledge from that domain (Kawahara, 2009). An interactive news navigator that generates dialogues based on news article archives has been developed along this concept (Yoshino et al., 2011).

The automatic speech recognition (ASR) module for spoken dialogue systems (SDSs) needs an appropriate language model (LM) adapted to the task domain and style. Even an ASR system with a very large vocabulary cannot cover all proper nouns or named entities (NEs), which are critical in information retrieval. Ideally, an LM should be trained with a large-scale matched corpus, but in many cases this is not realistic. Therefore, two approaches are commonly adopted. The first involves mixing document texts of the target domain with a dialogue corpus of spoken-style expressions. The other involves collecting relevant texts, possibly from spoken-style sentences, from the Web (Sarikaya et al., 2005; Sethy et al., 2005; Misu and Kawahara, 2006; Bulyko et al., 2007). These approaches try to cover the target domain and style of speech in an indirect way, but the resultant model will inevitably contain a large amount of irrelevant texts.

In general, information navigation systems with speech interfaces have a set of backend documents for retrieval (Schalkwyk et al., 2010). These systems require matching between the backend documents and the user utterance. Based on this assumption, we define a similarity measure of collected sentences from the Web (= expected user utterances) with the backend documents, and select well-matched sentences for LM training.

The overall flow of the proposed method is shown in **Figure 1**. In this paper, we assume two corpora: backend documents (D) and collected sentences (q) from the Web. Superficial matching, in which the method filters sentences based on the similarity in word sequences to the backend documents, is described in Section 2. N-gram model likelihood based on KL divergence is used in this method, and it is equivalent to the conventional perplexity-based method. In Section 3, we propose using deep semantic similarity based on P-A structures. This proposed method provides better filtering, considering not only surface words but also semantic cases. It is suitable forinformation navigation systems that require P-A structures. In Section 4, we propose the combined usage of the two filtering methods mentioned above. The method enables us to take both advantages of the two methods. We evaluate the conventional and proposed methods with ASR accuracy (word error rate (WER)), predicate-argument structure error rate (PAER), and test set perplexity in Section 5.

2 KL Divergence of N-gram Models

We construct an LM for ASR by using a collection of question-style queries (=sentences) from the Web, and matching them with the target backend documents D. For the matching of a question-style sentence q and backend documents D, first, we use KL divergence of LMs for surface word matching. KL divergence is a non-symmetric measurement of the difference

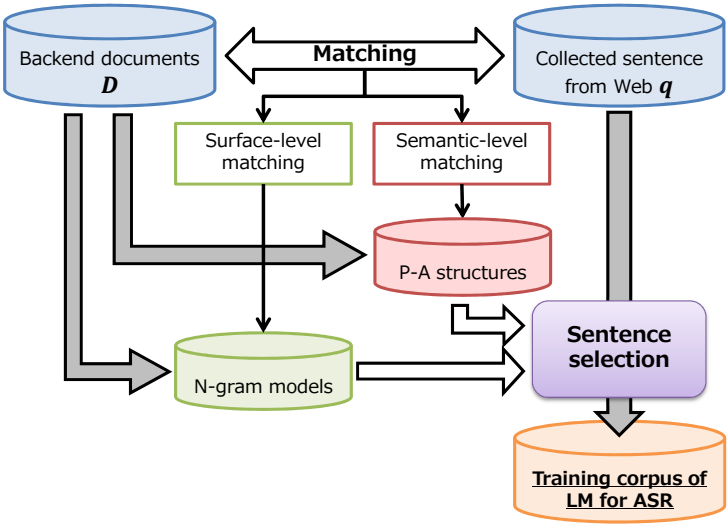

Figure 1: Overview of the proposed method

between two probability distributions (Kullback and Leibler, 1951). KL divergence of sentence q and backend documents D is defined as

$$KL(q||D) = \sum_m P_q(w_m) \log_2 \frac{P_q(w_m)}{P_D(w_m)}, \quad (1)$$

where $w_1 w_2, ..., w_n$ is the word sequence in sentence q. P_D and P_q are the probability distributions of D and q. We define these probability distributions with n-gram (tri-gram) models. We assume that query q consists of one sentence, and most of the time the n-gram distribution trained from only one sentence becomes unique. Then, the probability $P_q(W)$ becomes to 1. We can then make the approximation to KL divergence:

$$KL(q||D) \approx \sum_m \log_2 \frac{1}{P_D(w_m)} \quad (2)$$

$$= -\sum_m \log_2 P_D(w_m). \quad (3)$$

This formulation is equivalent to cross-entropy XE:

$$XE(q,D) = -\sum_m P_q(w_m) \log_2 P_D(w_m) \quad (4)$$

$$\approx -\sum_m \log_2 P_D(w_m). \quad (5)$$

The cross-entropy has two components: self information $P_q(w_m)$ and mutual information $P_D(w_m)$, but the self information $P_q(w_m)$ is equal to 1 as discussed above. These formulations

can be mapped to the definition of perplexity PP.

$$H(q,D) = -\frac{1}{n}\sum_{m=1}^{n}\log_2 P_D(w_m). \quad (6)$$

$$PP(q,D) = 2^{H(q,D)}. \quad (7)$$

There is already a method to filter sentences q based on the perplexity defined with targeted backend documents D (Misu and Kawahara, 2006). In this approach, we construct an n-gram model from the backend documents D and use the perplexity as a superficial similarity measure. This is equivalent to the KL divergence between q and D. We compute the perplexity $PP(q,D)$ to every collected sentence, and then rank them for the filtering.

3 Similarity based on Predicate-Argument Structure

In the conventional filtering method described in the previous section, the matching is performed using word-level similarity measure. However, it is difficult to select sentences that semantically match the backend documents because this method is based on surface information. This is problematic because information navigation systems work based on semantic structures of user utterance (Yoshino et al., 2011). We therefore propose a similarity measure based on P-A structures to select sentences that match the backend documents on the deep semantic level. In this section, our focus is on the semantic structures that are defined with P-A structures.

In the conventional approach, the similarity measure is defined with a generative model $P_D(w_i)$. In this section, we adopt a discriminative approach and calculate $P(D|w_i)$ to predict the significant sentences.

3.1 Predicate-Argument Structure

The P-A structure is automatically generated by a semantic parser (**Figure 2**). This P-A structure has a sub-structure that contains a predicate (w_p), argument (w_a), and its semantic case (w_s) (called a "P-A pair"). The P-A structures consist of various arguments that depend on one predicate with its semantic case. We used the JUMAN/KNP [1] analyzer to parse sentences and obtain the structures automatically. However, not every P-A pair is meaningful in information navigation; actually, only a fraction of the patterns are useful. For example, in the baseball domain, key patterns include "[A (agent) beat B (object)]" and "[A (agent) hit B (object)]", and in the business domain, "[A (agent) sell B (object)]" and "[A (agent) acquire B (object)]". The useful information structure is depending on the domain, and information extraction techniques have been investigated (Grishman, 2003). Conventionally, templates for information extraction have been hand-crafted (Ramshaw and Weischedel, 2005), but this heuristic process is so costly that it cannot be applied to the wide variety of domains on the Web. A method to automatically define domain-dependent templates for information extraction to be used in a flexible information navigation system has been proposed (Yoshino et al., 2011).

3.2 Significance Measure based on P-A structures

We extract important P-A pairs from the backend documents D and create matches between the extracted pairs and a question sentence q to measure the similarity based on the semantic

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

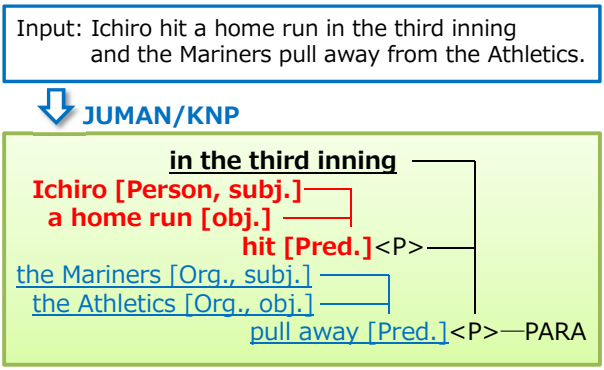

Figure 2: Example of predicate-argument (P-A) structure extraction.

significance. A previous study (Yoshino et al., 2011) has shown that an extraction method based on a Naive Bayes classifier is effective. In this method, the conditional probability of a document D (e.g., baseball) given a word w_i is defined as

$$P(D|w_i) = \frac{C(w_{i,D}) + x_D\gamma}{C(w_i) + \gamma}, \quad (8)$$

where γ is a smoothing factor estimated with a Dirichlet prior (Teh et al., 2006) using the Chinese Restaurant Processes (CRP). To calculate the conditional probabilities, we use sentences $\bar{D}$ of other domains that are extracted at random with the same population of D. x_D, a normalization factor that depends on the size of D, is defined as

$$x_D = \frac{\sum_j C(w_j, D)}{\sum_k C(w_k)}. \quad (9)$$

The P-A structure has a minimum sub-structure P-A pair (PA_i) containing the predicate (w_p), argument (w_a), and its semantic case (w_s). With the above definition, we define the conditional probability $P(D|PA_i)$ as

$$P(D|PA_i) = \sqrt{P(D|w_p, w_s) \times P(D|w_a)}. \quad (10)$$

3.3 Clustering of Named Entities

The statistical method often encounters the problem of data sparseness due to mismatch between the training set and the test set, especially with the named entities (NEs). To solve this problem, we cluster NEs that appear in the training set. NEs are one of the information structures that can be automatically generated by a semantic parser in accordance with a pre-defined category. An example of automatically labeled NEs is shown in the **Figure 2**, in which a labeler assigns person and organization labels to the entities.

We perform clustering to classify P-A structures that have the same trio of predicate, semantic case, and NE. In the example shown in **Figure 3**, two P-A structures that have the same abstract

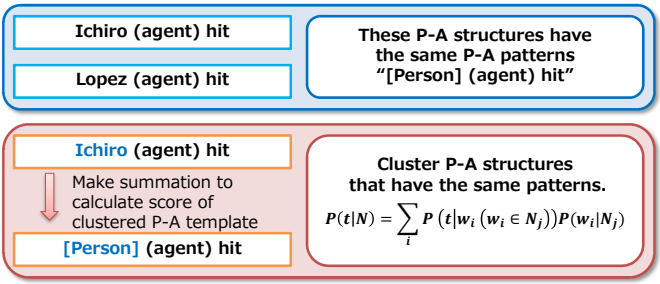

Figure 3: Clustering of named entities (NEs).

P-A pairs are clustered to the same template. We extend the probability of argument $P(w_a)$ as

$$P(D|N_i) = \sum_{k(w_k \in N_i)} P(D|w_k)P(w_k), \quad (11)$$

where N_i is the NE class that is included in arguments w_a. This clustering enables us to reduce lexicon mismatches between the backend documents D and question sentences q, leading to more robust matching.

3.4 Filtering with P-A Templates

For each question sentence q, we calculate the mean of every $P(D|PA_i)$ contained in the sentence q, which is defined as $P(D|q_{PA})$. An example of this scoring is shown in **Figure 4**. The input sentence q has four P-As. We take the mean of their scores to calculate $P(D|q_{PA})$.

Every sentence is ranked with $P(D|q_{PA})$, for selection to be used to train the LM for ASR. With this method, we can select sentences that are more relevant to the backend documents and more likely to be asked by users.

4 Combination of Sentence Selection Methods

We described two similarity measures for the backend documents D and question sentences q in Section 2 and 3. Considering their advantages, we propose a combination of the two methods using ranks and scores.

4.1 Method based on Sentence Rank

In this method, we sort question sentences q with the above-described $PP(q, D)$ and $P(D|q_{PA})$ and rank them as PP_{rank} and PA_{rank}. We then re-sort the sentences by summing the two ranks: $PP_{rank} + PA_{rank}$, for final selection.

4.2 A Method using Normalized Score

We re-define a new score by using $PP(q, D)$ and $P(D|q_{PA})$. The range of $P(D|q_{PA})$ is $0 < P(D|q_{PA}) < 1$, but the range of $PP(q, D)$ is not $0 < PP(q, D) < 1$, so we content it via the

P-A structure

q = "**Ichiro hit a game-winning double when the bases were loaded with two outs in the final inning.**"
PA = ["[Person]/subject/hit",
"a game-winning double/object/hit",
"the bases were loaded with two outs/locative/hit",
"final inning/modifier/hit"]

P-A templates

Score	Argument	case	Predicate	
0.99599	middle relievers		subject	lose
0.99519	relief pitcher		subject	lose
0.98716	final inning		modifier	hit
0.98202	a game-winning double		object	hit
0.98201	the bases were loaded with two outs		locative	hit
0.78062	[Person]		subject	hit
0.09994	share price		subject	slide
0.09994	charge		subject	increase
	...			

Figure 4: An example of $P(D|q_{PA})$ calculation.

sigmoid function.

$$PP' = \frac{1}{1 + e^{-PP}}. \tag{12}$$

A mixing ratio of 3:7 was set after a trial experiment.

5 Experimental Evaluation

We evaluated LMs constructed by the proposed methods with ASR. These LMs were constructed from selected sentences by using the four proposed filtering methods. We compared these models by using three indexes: word error rate (WER), predicate-argument structure error rate (PAER), and adjusted perplexity. PAER is based on the parsing accuracy of recognized sentences. Since the LMs have a different vocabulary size, we used adjusted perplexity, which penalizes smaller-vocabulary LMs.

5.1 Experimental Setting

We prepared an evaluation task using a news navigation system (Yoshino et al., 2011) in the professional baseball domain. Details of the test sets are shown in **Table 1**.

To train $PP(q, D)$ and $P(D|q_{PA})$, we prepared backend documents D of baseball articles from Mainichi newspaper articles (CD-Mainichi newspaper database 2000–2009). To train the LM for ASR, we used question-style sentences q taken from the baseball domain in the Yahoo! QA corpus [2] (a collection of queries on a Web site). The specification of the corpus are shown in **Table 2**. We used Julius[3] (Lee and Kawahara, 2001) as the ASR engine.

[2]This corpus was provided by Yahoo!JAPAN and National Institute of Informatics, Japan.
[3]http://julius.sourceforge.jp

Table 1: Specification of test sets.

Task	Users	Utterances
News navigation	10	2,747

Table 2: Specification of training sets.

Usage	Corpus	Sentences
Backend documents D	Mainichi newspaper articles	176,852
Pool of sentences q for training	Yahoo!QA entertainment: Baseball	403,602

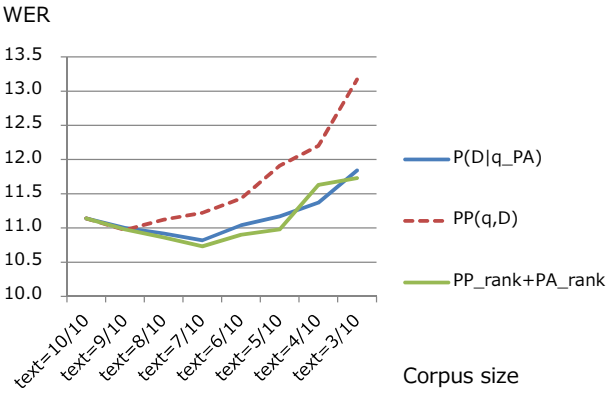

Figure 5: WER on news navigation task.

5.2 Experimental Results

The WER, PAER, and adjusted perplexity results are shown in **Figure 5, 6, and 7**, respectively. The horizontal axes are the relative size of the training set. There was a significant difference between the proposed method ($P(D|q_{PA})$, text $= 7/10$) and the baseline (text $= 10/10$, no filtering), with the significance level of less than 0.05 ($p < 0.05$.) There is no significant difference between the proposed method $P(D|q_{PA})$ and the combined $PP_{rank} + PA_{rank}$ rank or the combined $PP(q, D)$ and $P(D|q_{PA})$ score. Only the combined $PP_{rank} + PA_{rank}$ rank is shown in these graphs because there was not a distinguishable difference between the rank-based method and the normalized score-based method.

The experimental evaluation of WER shows that the similarity measure based on the semantic-level performed better than the conventional surface-level matching using n-gram models. The PAER evaluation also showed that the proposed method using a combination of ranks performed better than the conventional method. This demonstrates that using deep semantic-level similarity can improve the ASR accuracy.

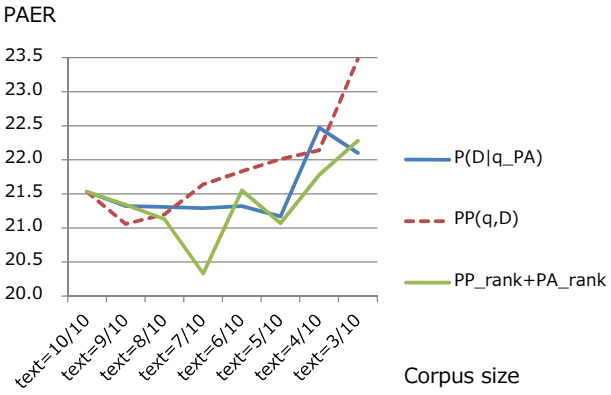

Figure 6: PAER on news navigation task.

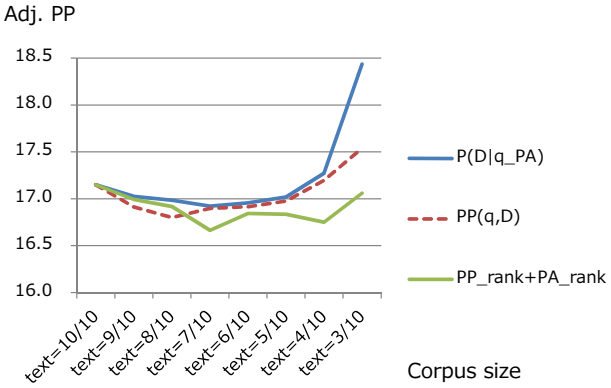

Figure 7: Adjusted perplexity on news navigation task.

6 Conclusion

We have proposed a method of sentence selection for language modeling for information navigation systems. The proposed method features sentence filtering based on semantic-level similarity. Experimental results showed that the proposed method performs better than the conventional method which uses only filtering based on surface-level perplexity. Moreover, the combinational usage of these two measures improve the ASR performance. Especially, filtering based on the P-A structure contributes the improvement of P-A structures accuracy (=reduce the PAER). As a future work, we plan to apply the method to a variety of domains.

References

Bulyko, I., Ostendorf, M., Siu, M., Ng, T., Stolcke, A., and Çetin, O. (2007). Web resources for language modeling in conversational speech recognition. *ACM Trans. Speech Lang. Process.*, 5(1):1:1–1:25.

Grishman, R. (2003). Discovery methods for information extraction. In *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pages 243–247.

Kawahara, T. (2009). New perspectives on spoken language understanding: Does machine need to fully understand speech? In *Proc. IEEE-ASRU*, pages 46–50.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Lee, A. and Kawahara, T. (2001). Julius–an open source real-time large vocabulary recognition engine. In *Proc. EuroSpeech*, pages 1691–1694.

Misu, T. and Kawahara, T. (2006). A bootstrapping approach for developing language model of new spoken dialogue system by selecting web texts. In *INTERSPEECH*, pages 9–13.

Misu, T. and Kawahara, T. (2010). Bayes risk-based dialogue management for document retrieval system with speech interface. *Speech Communication*, 52(1):61–71.

Ramshaw, L. and Weischedel, R. M. (2005). Information extraction. In *IEEE-ICASSP*, volume 5, pages 969–972.

Sarikaya, R., Gravano, A., and Gao, Y. (2005). Rapid language model development using external resources for new spoken dialog domains. In *Proc. ICASSP*, volume 1, pages 573–576.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Garret, M., and Strope, B. (2010). Google search by voice: A case study.

Sethy, A., Georgiou, P. G., and Narayanan, S. (2005). Building Topic Specific Language Models from Webdata Using Competitive Models. In *Proc. Interspeech*, pages 1293–1296.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Yoshino, K., Mori, S., and Kawahara, T. (2011). Spoken dialogue system based on information extraction using similarity of predicate argument structures. In *Proc. of SIGDIAL*, pages 59–66.