# A Semi-supervised Method for Efficient Construction of Statistical Spoken Language Understanding Resources

*Seokhwan Kim, Minwoo Jeong, and Gary Geunbae Lee*

Department of Computer Science and Engineering
Pohang University of Science and Technology, South Korea
{megaup,stardust,gblee}@postech.ac.kr

## Abstract

We present a semi-supervised framework to construct spoken language understanding resources with very low cost. We generate context patterns with a few seed entities and a large amount of unlabeled utterances. Using these context patterns, we extract new entities from the unlabeled utterances. The extracted entities are appended to the seed entities, and we can obtain the extended entity list by repeating these steps. Our method is based on an utterance alignment algorithm which is a variant of the biological sequence alignment algorithm. Using this method, we can obtain precise entity lists with high coverage, which is of help to reduce the cost of building resources for statistical spoken language understanding systems.

**Index Terms**: semi-supervised method, spoken language understanding

## 1. Introduction

During the past few years, spoken dialog systems have come into the limelight as more natural and convenient interfaces for users, and recently, the interest in spoken dialog systems has been sharply increasing. One of the main advantages of the spoken dialog systems is that the users can access information easily, even if they have no prior knowledge of using the system to obtain the information. In order to make this point effective, all the modules in spoken dialog systems should be trained with precise and up-to-date resources.

A spoken dialog system consists of several sub-modules, and each sub-module requires its own resources which are specified by the applied methodologies. Most of these resources have been built manually and have to be updated for reflecting up-to-date information, which causes high cost problem of building the system. In order to reduce the cost of building and maintaining the spoken dialog system which should be dealing with precise and up-to-date information, we propose a new semi-supervised information extraction method to be incorporated into the task of managing the sub-modules in spoken dialog systems. Existing works for semi-supervised information extraction commonly aim to automatically create context patterns with high-precision by integrating statistical characteristics of target texts with grammatical induction methodologies. Previous approaches mostly have dealt with the named-entity recognition problem [1][2][3][4], and some researchers have applied this framework to the relation extraction problem [5].

In this paper, we concentrate on utilizing our method for building resources for statistical spoken language understanding (SLU) modules in spoken dialog systems. The goal of SLU is to extract semantic meanings from recognized utterances and to fill the correct values into a semantic frame structure. Most of the statistical SLU approaches require a sufficient amount of training data which consist of labeled utterances with corresponding semantic concepts. Since the task of building the training data for the SLU problem is one of the most important and high-cost required tasks for managing the spoken dialog systems, we have incorporated our semi-supervised method in a semi-automatic procedure of building the training data for efficient prototyping of SLU systems.

Although an SLU problem can be considered as a kind of named-entity recognition problem and some well-developed techniques used for solving general NER problem, including semi-supervised approaches, can be applied to SLU problem, there exist several differences between characteristics of the two problems. Most of differences are caused by the fact that SLU problem deals with not literary style texts, but colloquial style spoken utterances. Spoken utterances are relatively free of grammar and have more noises as compared with literary text. Moreover, most people tend to speak in shorter and syntactically simplified utterances. These characteristics of spoken utterances cause lack of reliable context patterns indicating where the named-entities are located. Furthermore, the fact that it is difficult to obtain a large amount of data containing spoken utterances worsens the context sparseness problem. In order to solve this problem, we consider not only reliability, but also coverage of the method.

## 2. Semi-supervised Information Extraction for Spoken Language Understanding

The overall procedure of our semi-supervised information extraction method for SLU is operated as follows:

1. Prepare seed entity list E and unlabeled corpus C
2. Find utterances containing lexical of entities in E in the corpus C, and replace the parts of matched entities in the found utterances with a label which indicates the location of entities. Add partially labeled utterances to the context pattern set P.
3. Align each utterance in the corpus C with each context pattern in P, and extract new entity candidates in the utterance which is matched with the entity label in the context pattern.
4. Compute the score of extracted entity candidates in step 3, and add only high-scored candidates to E.
5. If there is no additional entities to E in step 4, terminate the process with entity list E, context pattern set P, and partially labeled corpus C as results. Otherwise, return to step 2 and repeat the process.

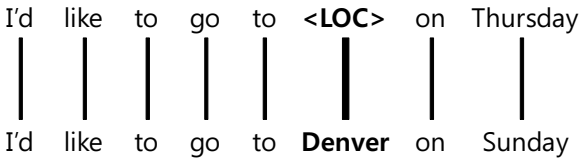


Figure 1: *Utterance alignment for named-entity recognition.*

### 2.1. Extracting Context Patterns

In order to obtain high-quality results of the context pattern induction method, the task of extracting reliable context patterns is more important than any other tasks in the method. The importance of the context pattern extraction is caused by the fact that just a few errors resulting from unreliable context patterns in an iteration may affect severely later iterations, and the phenomenon of error accumulation can bring about a serious performance loss of the overall procedure.

For general textual documents, a context pattern is a part of documents which indicates the existence and location of the entities, and it is mainly organized by a few phrases. On the other hand, it is not easy to obtain a sufficient amount of context patterns from spoken utterances in the same manner with the method for textual documents due to a lack of contextual information caused by relatively shorter and simpler fashion of most spoken utterances.

To overcome the context sparseness problem of spoken utterances, we make use of not sub-phrases of an utterance, but the full utterance itself as a context pattern for extracting named entities in the utterance. First, we assume that each entry in the entity list is absolutely precise and uniquely belong to only a category whether it is a seed entity or an extended entity as an intermediate of the overall procedure. For each entry in the entity list, we find out utterances containing it, and make an utterance template by replacing the part of entity in the utterance with the defined entity label. In this replacing task, we exclude the entities which are located at the beginning or end of the utterance, because context patterns containing the entities located in such positions can lead to confusion of determining the boundaries of each entity in the later procedure.

By using each utterance template itself as a context pattern to extract named-entities, we can obtain a reasonable amount of patterns easily. However, there also remains some problems on the quality of the extracted patterns because we blindly believe the precision of the entity list and do not consider the reliability or coverage of the extracted patterns. In our method, instead of sifting context patterns by scoring or ranking them, we preserve all the extracted patterns in this step and the works for solving the quality problems are attempted in the next step based on a pairwise utterance alignment method.

### 2.2. Pairwise Alignment

As stated above, we treat an utterance template as a context pattern, while previous researchers used simple phrasal rules [1] [2] [3] or induced automata [4] as a context pattern to extract named-entities. Due to the difference of the unit of context patterns from the previous approaches, we cannot simply apply the previously reported methods of utilizing context patterns to extract named-entities, hence we should devise a new extraction method. For extracting named-entities based on our proposed utterance template-based context patterns, we propose an utterance alignment method. As shown in Fig.1, we firstly align a raw utterance with a context pattern containing entity labels. Then, from the result of the best alignment between them, we extract the parts of the raw utterance which are aligned to the entity labels in the context pattern as an entity candidate belonging to the category of the corresponding entity label. In Fig.1, 'Denver' is extracted as an entity candidate of the location category. Using the utterance alignment method, we can obtain the alignment results for all pairs between each context pattern and each raw utterance, as well as for the extended entities from the alignment results with high alignment scores.

| | I'd | like | to | go | to | <LOC> | on | Thursday |
|---|---|---|---|---|---|---|---|---|
| I'd | 1 | | | | | | | |
| like | | 1 | | | | | | |
| to | | | 1 | | | | | |
| go | | | | 1 | | | | |
| to | | | | | 1 | | | |
| Denver | | | | | | 0 | | |
| on | | | | | | | 1 | |
| Sunday | | | | | | | | 0 |

| | I'd | like | to | go | to | <LOC> | on | Thursday |
|---|---|---|---|---|---|---|---|---|
| I'd | 1 | | | | 1 | 1 | 0 | 0 |
| like | | 1 | | | 1 | 1 | 0 | 0 |
| to | | | 1 | | 1 | 1 | 0 | 0 |
| go | | | | 3 | 1 | 1 | 0 | 0 |
| to | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 0 |
| Denver | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| on | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Sunday | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 2: *Matrix initialization and filling steps for utterance alignment.*

In order to achieve high performance of the overall method, the problem of how we can reliably align a raw utterance with the context pattern has to be dealt with as the most important part of the method. To solve this problem, we present an utterance alignment method based on the Needleman-Wunsch algorithm[6]. The Needleman-Wunsch algorithm is widely used in bioinformatics to align nucleotide or protein sequences. We utilize this algorithm into the utterance alignment task by considering a word as a unit of alignment instead of biological residues.

The alignment algorithm is performed by computing the score for each word pair in the alignment matrix. Each column in the matrix corresponds to a word in the context pattern, while each row in the matrix corresponds to a word in the raw utterance. Moreover, a point of crossing between a row and a column has the score of aligning the word in the raw utterance with the word in the context pattern.

The first step in the utterance alignment method is to assign the initial value of each position in the matrix $M$ with

$$M_{i,j} = \begin{cases} 1 & \text{if } tar(i) \text{ and } ref(j) \text{ are identical} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $tar(i)$ is the $i$-th word in the raw utterance and $ref(j)$ is the $j$-th word in the context pattern. Fig.2(a) shows an example of initialization step.

In the matrix filling step, we find the maximum alignment score by starting in the lower right hand corner in the matrix and finding the maximal score $M_{i,j}$ for each position in the matrix. The maximum score $M_{i,j}$ is defined as

$$M_{i,j} = M_{i,j} + \max \begin{pmatrix} M_{i+1,j+1} \\ \max_{k=j+2}^{n}(M_{i+1,k}) \\ \max_{k=i+2}^{m}(M_{k,j+1}) \end{pmatrix}, \quad (2)$$

where $n$ is the number of words in the context pattern and $m$ is the number of words in the raw utterance.

For example, in Fig.2(b),

$$\begin{aligned} M_{4,4} &= M_{4,4} + \max \begin{pmatrix} M_{5,5} \\ \max_{k=6}^{8}(M_{5,k}) \\ \max_{k=6}^{8}(M_{k,5}) \end{pmatrix} \\ &= 1 + \max\left(2, \max\left(1,0,0\right), \max\left(1,0,0\right)\right) \\ &= 3. \end{aligned} \quad (3)$$

The values of other positions are computed in the same manner, and the matrix $M$ is filled in the direction of the upper left hand corner in the matrix as shown in Fig.3. After the matrix filling step, we trace back the matrix to find the best alignment with the maximum score and extract the entity candidates from the result of alignment.

The traceback step is started at the position with maximum score from among the first column and the first row. Then, the next position of the position $[i, j]$ is determined by following policies.

- If $tar(i)$ and $ref(j)$ are identical, then the next position is $[i + 1, j + 1]$.
- Otherwise, the position with maximum score from among $[i + 1, j + 1] \sim [i + 1, n]$ and $[i + 1, j + 1] \sim [m, j + 1]$ is the next position.

In Fig.3, the sequence of positions with gray color indicates the best alignment with maximum alignment score. From the result of the alignment, we can extract the word 'Denver' as an entity candidate for the 'location' category.

Assuming that the alignment algorithm is impeccable, the overall procedure seems to be reasonable. However, although the Needleman-Wunsch algorithm guarantees to find the alignment with the maximum alignment score, there exists a critical problem in the method. The problem is that the method cannot extract multi-word entities, because the algorithm is aligned for each pair of the single units only.

In order to make it possible to handle multi-word entities, we modified the algorithm by revising the traceback policy. We append a following policy into the existing policies.

- If $ref(j)$ is an entity label and $tar(i)$ is not, then the position with maximum score from among $[i + 1, j + 1] \sim [i + 1, n]$, $[i + 1, j + 1] \sim [m, j + 1]$ and $[i + 1, j]$ is the next position.

Since considering the position of $[i + 1, j]$ additionally for the case that the $j$-th word in the context pattern is an entity label, we can extract not only single-word entities, but also multi-word entities, as shown in Fig.4.

| | I'd | like | to | go | to | <LOC> | on | Thursday |
|---|---|---|---|---|---|---|---|---|
| I'd | **6** | 4 | 3 | 2 | 1 | 1 | 0 | 0 |
| like | 4 | **5** | 3 | 2 | 1 | 1 | 0 | 0 |
| to | 3 | 3 | **4** | 2 | 1 | 1 | 0 | 0 |
| go | 2 | 2 | 2 | **3** | 1 | 1 | 0 | 0 |
| to | 1 | 1 | 1 | 1 | **2** | 1 | 0 | 0 |
| **Denver** | 1 | 1 | 1 | 1 | 1 | **1** | 0 | 0 |
| on | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
| Sunday | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |

Figure 3: *Traceback step for utterance alignment and named-entity recognition.*

| | I'd | like | to | go | to | <LOC> | on | Thursday |
|---|---|---|---|---|---|---|---|---|
| I'd | **6** | 4 | 3 | 2 | 1 | 1 | 0 | 0 |
| like | 4 | **5** | 3 | 2 | 1 | 1 | 0 | 0 |
| to | 3 | 3 | **4** | 2 | 1 | 1 | 0 | 0 |
| go | 2 | 2 | 2 | **3** | 1 | 1 | 0 | 0 |
| to | 1 | 1 | 1 | 1 | **2** | 1 | 0 | 0 |
| **New** | 1 | 1 | 1 | 1 | 1 | **1** | 0 | 0 |
| **York** | 1 | 1 | 1 | 1 | 1 | **1** | 0 | 0 |
| on | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
| Sunday | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |

Figure 4: *An example of multi-word named-entity recognition.*

### 2.3. Scoring Entity Candidates

Since we do not consider the reliability of the context patterns so far, how to assign the score to each extracted entity candidate is more important. Before defining the score for each entity, we define the score of the alignment between a raw utterance and a context pattern as

$$score(ref, tar) = \frac{2 \cdot M_{0,0}}{(n - t) + (m - e)}, \quad (4)$$

where $ref$ is a context pattern, $tar$ is a raw utterance, $n$ is the number of words in $ref$, $m$ is the number of words in $tar$, $t$ is the number of aligned entity labels, and $e$ is the number of words extracted as entity candidates.

By using the defined alignment score, we define the score of an entity candidate $e_j$ which is extracted by a context pattern $ref_i$ as

$$score(e_j, ref_i) = \frac{\sum_{tar_k \ni e_j} score(ref_i, tar_k)}{\text{\# of } tar_k \ni e_j}. \quad (5)$$

Also, we define the final score of an entity candidate $e_j$ as

$$score(e_j) = \frac{\sum_i score(e_j, ref_i)}{\text{\# of context patterns}}, \quad (6)$$

which has the value between 0 and 1, and higher means more reliable.

Table 1: *Result of automatic entity list extension*

| Category | # of seeds | # of extended entities | # of total entities | # of iterations | Precision | Recall |
|---|---|---|---|---|---|---|
| CITY_NAME | 20 | 123 | 209 | 4 | 65.04% | 37.91% |
| MONTH | 1 | 10 | 12 | 2 | 100% | 83.33% |
| DAY_NUMBER | 3 | 27 | 34 | 3 | 100% | 79.41% |

## 3. Experiments

We evaluated our method on the CU-Communicator corpus, which consists of 13,983 utterances. We chose the three most frequently occurring semantic categories in the corpus, CITY_NAME, MONTH, and DAY_NUMBER, and filtered out the other categories which occurred relatively rarely.

For each category, we randomly selected one-tenth of the entities as seed entities from the labeled entities in the corpus. Then, we started the experiment with these seed entities and raw utterances which are obtained by removing labels from the original corpus. Repeating by iteration, some entities are extracted as entity candidates. From these candidates, we selected candidates with higher scores than a threshold value, and each selected candidate is appended to the entity list of the corresponding category. To reduce the propagated errors, we empirically set the entity selection threshold value to 0.3. We evaluated the quality of extended entity lists by measuring precision and recall to the labeled entity lists in the original corpus. The experimental results are shown in Table 1. For MONTH and DAY_NUMBER categories, we obtained perfectly precise entity lists with reasonable coverage, which is expected for the closed set category with only a few elements, such as $E_{month} = \{January, February, ..., December\}$.

On the other hand, the result on the open set category such as CITY_NAME category indicates that the extended entities on the category are relatively imprecise and cannot include the majority of the existing entities in the corpus. From the analysis on the errors, we discovered that 53.5% of the errors include not only CITY_NAME, but also STATE_NAME, such as "Minneapolis Minnesota". In the case that a STATE_NAME is followed right after a CITY_NAME, it is difficult to extract the CITY_NAME only, because there is no punctuation mark on spoken utterances to separate them. But these errors are not critical in most of the SLU tasks. Then, we found out that 96.2% of the undetected entities on the corpus actually occurred less than 10 times. These rarely occurred entities with low possibilities to be extracted by a sufficient number of context patterns can decrease the recall of existing entities, but cannot much affect to the following final result for corpus labeling, because of their low frequency.

Owing to the assumption that each entry in the entity list uniquely belongs to only one category, we can utilize the extended entity lists to make a prototype of the training data for other supervised methodologies. We discovered the parts of raw utterances which are identical to the entries in the extended entity list, and labeled them to the corresponding labels. We obtained a labeled corpus with an F-score of 89.22, as shown in Table. 2. From the result, we conclude that our method can reduce the cost of labeling corpus for SLU to about one-tenth of manual labeling, while maintaining the high performance.

Table 2: *Result of corpus labeling experiment*

| Category | Precision | Recall | F-measure |
|---|---|---|---|
| CITY_NAME | 91.30 | 86.83 | 89.01 |
| MONTH | 98.98 | 87.24 | 92.74 |
| DAY_NUMBER | 92.00 | 82.03 | 86.73 |
| Overall | 93.24 | 85.53 | 89.22 |

## 4. Conclusion

We have presented a semi-supervised information extraction method which is based on a modified sequence alignment algorithm. Using our method, we obtained high quality labeled corpus starting with just a few seed entities. However, it leaves much room for improvement on our method. In this paper, we did not consider the similarity values and gap penalty which are essentially considered in the original sequence alignment algorithm. If we utilize them as an effective means of reflecting more information, then we can obtain much better results with much lower cost. Refining the method and applying it to other problems such as back-end knowledge expansion for spoken dialog systems is a topic of our future works.

## 5. Acknowledgements

This research was supported by the Invitation for Proposals on Internet Services Theme funded by Microsoft Research Asia, and also supported as a Brain Neuroinformatics Research Program sponsored by Ministry of Commerce, Industry and Energy.

## 6. References

[1] Riloff, E. and Jones, R., "Learning Dictionaries for Information Extraction by Multi-level Bootstrapping", In Proceedings of the Sixteenth National Conference on Artificial Intelligence. 1999.

[2] Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A., "Unsupervised named-entity extraction from the web - an experimental study", Artificial Intelligence Journal. 2005.

[3] Lee, S. and Lee, G. G., "Exploring phrasal context and error correction heuristics in bootstrapping for geographic named entity annotation", Information Systems. 32(4):575-592. 2007.

[4] Talukdar, P. P., Brants, T., Liberman, M., and Pereira, F., "A Context Pattern Induction Method for Named Entity Extraction", In Procedings of the 10th Conference on Computational natural Language Learning (CoNLL-X). 2006.

[5] Agichtein, E. and Gravano, L., "Snowball: Extracting relations from large plain-text collections", In Proceedings of the Fifth ACM International Conference on Digital Libraries. 2000.

[6] Needleman, S. B. and Wunsch, C. D., "A general method applicable to the search for similarities in the amino acid sequence of two proteins", J. Mol. Biol. 48(3):443-53. 1970.