Cross-Language Latent Relational Search: Mapping Knowledge across Languages

Nguyen Tuan Duc
duc@mi.ci.i.u-tokyo.ac.jp

Danushka Bollegala
danushka@iba.t.u-tokyo.ac.jp

Mitsuru Ishizuka
ishizuka@i.u-tokyo.ac.jp

The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract

Latent relational search (LRS) is a novel approach for mapping knowledge across two domains. Given a source domain knowledge concerning the Moon, "The Moon is a satellite of the Earth", one can form a question {(Moon, Earth), (Ganymede, ?)} to query an LRS engine for new knowledge in the target domain concerning the Ganymede. An LRS engine relies on some supporting sentences such as "Ganymede is a natural satellite of Jupiter." to retrieve and rank "Jupiter" as the first answer. This paper proposes cross-language latent relational search (CLRS) to extend the knowledge mapping capability of LRS from cross-domain knowledge mapping to cross-domain and cross-language knowledge mapping. In CLRS, the supporting sentences for the source pair might be in a different language with that of the target pair. We represent the relation between two entities in an entity pair by lexical patterns of the context surrounding the two entities. We then propose a novel hybrid lexical pattern clustering algorithm to capture the semantic similarity between paraphrased lexical patterns across languages. Experiments on Japanese-English datasets show that the proposed method achieves an MRR of 0.579 for CLRS task, which is comparable to the MRR of an existing monolingual LRS engine.

Introduction

Latent relational search (Kato et al. 2009; Duc, Bollegala, and Ishizuka 2010; Goto et al. 2010) is a novel search paradigm based on the proportional analogy between two entity pairs. When the relation between two entities A and B is highly similar to that between two entities C and D, we say that the entity pair (A, B) and (C, D) have a high degree of *relational similarity* (Turney 2005). Given a latent relational search (LRS) query {(A, B), (C, ?)}, an LRS engine is expected to retrieve an entity D as an answer to the question mark (?) in the query, such that the pair (A, B) and (C, D) have a high degree of relational similarity. For example, the answer for the query {(Moon, Earth), (Ganymede, ?)} is expected to be "Jupiter", because the relation between Moon and Earth is highly similar to that between Ganymede and Jupiter (The Moon is a natural satellite of the Earth, whereas, Ganymede is a natural satellite of Jupiter). In LRS, our knowledge of a familiar domain (the Moon in the above example) has been mapped to an unfamiliar domain (the Ganymede) to discover new knowledge. Therefore, LRS is useful for knowledge discovery in an unfamiliar domain.

This paper proposes *cross-language latent relational search* (CLRS) to extend the knowledge mapping capability of latent relational search from cross-domain knowledge mapping to cross-domain and cross-language knowledge mapping. In CLRS, the input entity pair might be in a different language with the target pair. Moreover, the supporting sentences for the input pair might be in a different language from those of the target pair. For example, given the Japanese-to-English query {(月, 地球), (Ganymede, ?)} (meaning {(Moon, Earth), (Ganymede, ?)}), a CLRS engine is expected to output "Jupiter" as an answer and to retrieve supporting sentences in Japanese for the input pair and in English for the target pair.

Following previous work on monolingual relational similarity measuring (Turney 2005; Bollegala, Matsuo, and Ishizuka 2009), we represent the relation between two entities by lexical patterns of the context surrounding the two entities. We then propose a hybrid (soft/hard) lexical pattern clustering algorithm to recognize paraphrased lexical patterns across languages. Using the result of the proposed clustering algorithm, we can precisely measure the relational similarity between two entity pairs that are in different languages and can therefore precisely rank the result list of a CLRS query. We improve the precision by combining the proposed clustering algorithm with cross-language Latent Relational Analysis (LRA) (Turney 2005). When evaluating with a 1.6GB Japanese-English web corpus, the proposed method achieves an MRR of 0.579, which is better than the MRR of a previous monolingual latent relational search system.

Method

Cross-language entity pair and relation extraction

Given a text corpus containing documents of several languages (but not necessary a parallel or aligned corpus), we first pre-process the corpus to create an index for high speed retrieval. The index that we create is a multi-lingual extension of the index in (Duc, Bollegala, and Ishizuka 2010).

The index keeps information concerning entity pairs and lexical patterns in multiple languages. Before extracting entity pairs and relations from a document in the corpus, we identify the language of the document. There are many fast algorithms for identifying language of a document with high accuracy (McNamee 2005). Because in our experiments, we only need to classify between Japanese and English web pages, we simply count the number of Japanese characters in a document. If this number is greater than 5% of the total number of characters, we assume that the document is a Japanese document. We then split the document into sentences using a sentence boundary detector for the identified language. From each sentence, we use the algorithm in (Duc, Bollegala, and Ishizuka 2010) to extract all named entity pairs [1] and lexical patterns that might represent the semantic relation between two entities in each pair. This algorithm allows extracting discontinuous fragments from the context surrounding an entity pair. For example, from the sentence "While no law stated that Tokyo is the capital of Japan, many laws define the 'capital area', which contains the Tokyo Metropolis.", the algorithm extracts three entity pairs (Tokyo, Japan), (Japan, Tokyo Metropolis) and (Tokyo, Tokyo Metropolis). To extract lexical patterns for the pair (Tokyo, Japan), the algorithm first replaces *Tokyo* with the variable X and *Japan* with the variable Y to make the lexical patterns independent from the entity pair. It then extracts discontinuous fragments (which are connected by asterisks "$*$") from the text window surrounding the pair, such as "stated that X is the capital of Y, many laws", "X is the capital of Y", "X * capital * Y", "X * capital of Y" (the asterisk "$*$" means zero or more words). Because the algorithm extracts discontinuous fragments from the text window, we have some common lexical patterns from two entity pairs even when the text in the gap between each pair is not match. For example, from the sentence "Pretoria is the de facto capital of South Africa", the algorithm can extract the lexical pattern "X * capital of Y" for the pair (Pretoria, South Africa). This allows the pair (Pretoria, South Africa) to have many common lexical patterns with the pair (Tokyo, Japan) in the previous sentence. Therefore, the relational similarity between the two pairs is high.

We create a lexical pattern vs. entity pair co-occurrence matrix $\mathbf{M}$ whose rows correspond to lexical patterns and columns correspond to entity pairs. The element $\mathbf{M}_{ij}$ of $\mathbf{M}$ is the number of co-occurrences between the i^{th} lexical pattern and the j^{th} entity pair. It is worth noting that, the matrix $\mathbf{M}$ conveys co-occurrence information of pairs and lexical patterns in multiple languages.

Entity pair and lexical pattern translation

Although we can associate some parallel lexical patterns (which are the translations of each other) with a same entity pair during the cross-language indexing process in previous section, the number of these associations might be very small. Therefore, we increase the number of known parallel patterns by using machine translation to find reliable parallel patterns. To find the parallel pattern of a lexical pattern, we first replace the variables X and Y in the pattern with popular entity pairs that a machine translation system can easily identify the translation of the entities in the target language. For example, if the NE tag of X is LOCATION, then we replace X with Tokyo. We then use a statistical machine translation (SMT) system [2] to translate the pattern into the target language. From the translation result, we identify the position of the two entities in the original pattern and replace them with X and Y. We verify the translation result by looking up the translated pattern in the index. If we can actually find the translated pattern in the index, then there is a high probability that the SMT system has produced a good result. Therefore, we mark the input pattern and the translated pattern as parallel patterns. We do the same for entity pairs to find parallel entity pairs (i.e., entity pairs that are the translations of each other). We then merge two columns of the matrix $\mathbf{M}$ that correspond to two parallel entity pairs into one. Similarly, we merge two rows that correspond to two parallel lexical patterns into one. After translation and merging, we obtain a co-occurrence matrix $\mathbf{A}$ (with smaller size than $\mathbf{M}$) in which parallel patterns and entity pairs are merged together. We can then use this matrix to measure the cosine similarity between two rows or two columns. We denote the cosine between two rows i and j in the matrix $\mathbf{A}$ (corresponding to two lexical patterns p_i and p_j) as $\mathrm{sim}_{\mathrm{VSM}}(p_i, p_j)$, as frequently used in the Vector Space Model (VSM).

Cross-language Latent Relational Analysis

To compress similar entity pairs into one dimension while measuring the similarity between two lexical patterns, we apply Latent Relational Analysis (LRA) (Turney 2005) to the matrix $\mathbf{A}$. Because $\mathbf{A}$ contains co-occurrence information of pairs and patterns in multiple languages, this process can be viewed as cross-language LRA. (Bollegala, Matsuo, and Ishizuka 2010) suggest that we should filter very rare patterns and entity pairs before LRA or clustering. This is because rare patterns are normally noisy patterns, which frequently contain strange symbols or misspellings; rare entity pairs are often caused by errors of the Named Entity Recognizers.

LRA uses Singular Value Decomposition (SVD) to decompose the matrix $\mathbf{A}$ into three matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{1}$$

The matrix $\mathbf{\Sigma}$ is a rectangular diagonal matrix whose diagonal elements are singular values of $\mathbf{A}$ (Turney 2005; Manning, Raghavan, and Schutze 2008). We can reduce the dimension of the row vectors of $\mathbf{A}$ by taking only the k biggest singular values in the matrix $\mathbf{\Sigma}$ (we denote this reduced matrix as $\mathbf{\Sigma}_k$). We can then measure the similarity between two lexical patterns in the reduced vector space by taking the cosine between two corresponding rows in the matrix $\mathbf{U}_k\mathbf{\Sigma}_k$, where $\mathbf{U}_k$ is the reduced matrix of $\mathbf{U}$. We denote the cosine similarity between two lexical patterns in the reduced space as $\mathrm{sim}_{\mathrm{LRA}}(p_i, p_j)$.

[1] We use the Stanford and the MeCab POS/NE tagger
http://nlp.stanford.edu/software/CRF-NER.shtml
http://mecab.sourceforge.net/

[2] http://translate.google.com/

Hybrid lexical pattern clustering algorithm

The matrix **A** compresses some parallel (or similar) lexical patterns into one dimension but it can only discover a very small number of similar patterns. This is because a semantic relation can be stated in several ways in natural language. For example, the lexical patterns "X acquired Y" and "X purchased Y" represent similar semantic relation. Likewise, the patterns "*X ga Y wo baishu shita*" and "*X ni yoru Y no baishu*" represent the acquisition relation in Japanese (we write these patterns using the English alphabet for convenience). Therefore, we need to recognize these paraphrased patterns and compress them into one dimension in order to precisely measure the relational similarity between two entity pairs. (Bollegala, Matsuo, and Ishizuka 2009) propose a sequential clustering algorithm to solve this problem in monolingual case. For each pattern, the algorithm finds the most similar cluster (that has been created so far) to the pattern. If the similarity between the pattern and the cluster is higher than a clustering similarity threshold θ then the pattern is added to the cluster, otherwise it forms a new singleton cluster. However, this algorithm might not work well for multi-lingual case. This is because a lexical pattern that co-occurs with entity pairs of multiple languages normally has smaller similarity with a pattern that co-occurs with entity pairs of only one language. For example, the pattern "X purchased Y" co-occurs with only English entity pairs. However, the merged pattern {"X acquired Y", "*X ga Y wo baishu shita*"} co-occurs with both English and Japanese entity pairs. Therefore, the $\mathrm{sim}_{\mathrm{VSM}}$ between "X purchased Y" and "*X ga Y wo baishu shita*" will be small. This implies that the clustering similarity threshold θ should not be uniform for all patterns in the clustering algorithm.

To solve this problem, we propose a two-phase sequential clustering algorithm to recognize paraphrased lexical patterns in a same language and across languages, as shown in Algorithm 1. In the first phase of our clustering algorithm, we want to capture the semantic similarity between paraphrased lexical patterns in the same language. Therefore, we use the sequential pattern clustering algorithm in (Bollegala, Matsuo, and Ishizuka 2009) in this phase. First, it sorts the pattern set in the order of frequency from high to low to process high frequency patterns first. For each pattern, the algorithm finds the cluster whose centroid has maximum similarity with the pattern (line 5). The similarity between two patterns p_i, p_j can be calculated using $\mathrm{sim}_{\mathrm{VSM}}$ or $\mathrm{sim}_{\mathrm{LRA}}$ (as defined above). If the similarity is above a pattern clustering similarity threshold θ_1 then the pattern is added to the cluster, otherwise, the pattern forms a new singleton cluster itself (line 10–15). Therefore, this algorithm is a hard clustering algorithm (i.e., each pattern can be in only one cluster).

In the second phase, we use a soft clustering algorithm with a lower pattern clustering similarity threshold θ_2 to associate parallel patterns to the pattern clusters that we obtained in the first phase. That is, we consider only patterns that have some parallel partners for clustering in the second phase (line 19 of the Algorithm 1), and we allow each of these patterns to be associated with many pattern clusters. If the similarity between a pattern that has some parallel partners and the centroid of a pattern cluster is above θ_2, we add the pattern and its parallel partners to the cluster (line 20–24). We need to associate as many parallel patterns as possible to these clusters to increase the recall as well as the precision of cross-language queries. A soft clustering algorithm in this phase accomplishes this goal, because a pattern and its parallel partners are allowed to appear in multiple clusters.

Algorithm 1 Hybrid Sequential Clustering (HSC) of lexical patterns

Input: pattern set $\wp$, threshold $\theta_1 > \theta_2 > 0$
Output: cluster set $\mathbf{K}$
1: $\mathbf{K} \leftarrow \{\}$
2: // First phase
3: sort($\wp$)
4: **for** pattern $p \in \wp$ **do**
5: $\quad maxClus \leftarrow \mathrm{argmax}_{c \in \mathbf{K}} \mathrm{sim}(p, \mathrm{centroid}(c))$
6: $\quad maxSim \leftarrow -1$
7: $\quad$ **if** $maxClus \neq \mathrm{NULL}$ **then**
8: $\quad\quad maxSim \leftarrow \mathrm{sim}(p, \mathrm{centroid}(maxClus))$
9: $\quad$ **end if**
10: $\quad$ **if** $maxSim \geq \theta_1$ **then**
11: $\quad\quad maxClus.\mathrm{append}(p)$
12: $\quad$ **else**
13: $\quad\quad newClus \leftarrow \{p\}$
14: $\quad\quad \mathbf{K} \leftarrow \mathbf{K} \cup \{newClus\}$
15: $\quad$ **end if**
16: **end for**
17: // Second phase
18: **for** pattern $p \in \wp$ **do**
19: $\quad$ **if** $\mathrm{hasParallel}(p)$ **then**
20: $\quad\quad$ **for** cluster $c \in \mathbf{K}$ **do**
21: $\quad\quad\quad$ **if** $\mathrm{sim}(p, \mathrm{centroid}(c)) \geq \theta_2$ **then**
22: $\quad\quad\quad\quad c.\mathrm{append}(p)$
23: $\quad\quad\quad\quad c.\mathrm{append}(\mathrm{paralellOf}(p))$
24: $\quad\quad\quad$ **end if**
25: $\quad\quad$ **end for**
26: $\quad$ **end if**
27: **end for**
28: **return K**

The second phase is an important step in our algorithm because it captures the semantic similarity between patterns in different languages. Even when two patterns in two different languages share only a small number of entity pairs so that they failed to be in a cluster in the first phase (because the similarity is much lower than θ_1), they can be grouped in a same cluster in the second phase, because of the low similarity threshold value. Therefore, the second phase is mainly to capture the similarity between translated (or paraphrased) patterns across languages.

Retrieving and ranking candidates

Given the query {(A, B), (C, ?)}, we first enumerate all lexical patterns of the input pair $s = (A, B)$. We then list all pairs of the form (C, X) which have at least one lexical pattern in the same cluster with a pattern of the input

Table 1: Relation types to retrieve a test corpus (italic entities are actually in Japanese)

Relation type	Example
BIRTHPLACE	(Franz Kafka, Prague), (*Hamasaki Ayumi*, *Fukuoka*), ...
HEADQUARTERS	(Google, Mountain View), (*Toyota*, *Aichi*) ...
CEO	(Eric Schmidt, Google), (*Toyoda Akio*, *Toyota*), ...
ACQUISITION	(Google, Youtube), (*Panasonikku*, *Sanyo*), ...
PRESIDENT	(Barack Obama, U.S), (*Sarukozi*, *Furansu*) ...
PRIMEMINISTER	(David Cameron, U.K), (*Kan Naoto, Nihon*) ...
CAPITAL	(Paris, France), (*Tokyo, Nihon*) ...
SATELLITE	(Ganymede, Jupiter), (*Oberon, Tennosei*)

pair. Because lexical patterns in a cluster often have similar meaning, if the pair $c = (C, X)$ has lexical patterns in the same cluster with lexical patterns of (A, B), then there is a high probability that (C, X) is relationally similar to (A, B). Therefore, we use the above list as the candidate list. We then measure the relational similarity between s and c using the cosine between two column vectors of the matrix **A** that correspond to s and c. When calculating the cosine, we assume that two lexical patterns that are in the same pattern cluster are equal (i.e., compressing them into one dimension). We rank the candidate list using this relational similarity.

Experiments

Data set

We evaluate our system with eight types of relations, as shown in Table 1. These relation types are frequently used in previous research to evaluate relational similarity measuring algorithm (Bollegala, Matsuo, and Ishizuka 2009), monolingual latent relational search engines (Kato et al. 2009; Duc, Bollegala, and Ishizuka 2010) or relation extraction systems (Banko et al. 2007; Bunescu and Mooney 2007). To create a corpus that includes Web pages concerning these semantic relations, we query Google[3] to retrieve the Top 100 URLs that are relevant to each entity pair. From the URL set, we crawl the HTML page at each URL. After the crawling process, we obtain a training dataset (1.8GB) and a test dataset (1.6GB) containing Web pages in Japanese and English. These sets of web pages contain a large number of entities and relations of many types (not only those in Table 1 because a web page might describe many entities and relations). We then run the pre-processing phases on the generated text corpus to build the index for our system.

We create 16 query sets[4] to evaluate the system, eight query sets are English-to-Japanese query sets, the other eight sets are Japanese-to-English. Each query set corresponds to a relation type in Table 1 and contains 50 queries. A set of 50 queries is large enough to evaluate the performance of an IR system (Manning, Raghavan, and Schutze 2008). Each query has only one correct answer. For example, we create the query {(?, YouTube), (*Panasonikku*, *Sanyo*)} for the ACQUISITION relation and {(Ganymede, Jupiter), (*Oberon*, ?)} for the SATELLITE relation. The criteria for evaluation is the Mean Reciprocal Rank (MRR) of each query set. MRR reflects both recall and precision of a search engine and is frequently used for evaluating search engines (Manning, Raghavan, and Schutze 2008).

For convenience, if we use $\mathrm{sim}_{\mathrm{VSM}}$ in the Algorithm 1 to calculate the similarity between two lexical patterns then we call the method as **HSC** (hybrid sequential clustering). If we use $\mathrm{sim}_{\mathrm{LRA}}$ then we call the method as **HSC+LRA**.

Parameter tuning

In these experiments, we evaluate the system using four relation types in the first four rows of Table 1. The performance is the average performance on eight query sets (four English-to-Japanese sets) corresponding to these four relation types. We run the proposed extraction algorithm on the training dataset (1.8GB corpus) to build an index for the system. The resulting index contains 5,241,627 lexical patterns and 236,923 entity pairs. Only 149,835 patterns that do not contain the wildcard character ("$*$") are considered for translation. After the pattern translation process, we found 6812 patterns that have parallel patterns. Therefore, the ratio of reliable translation is only 4.55% and only 0.27% of the total number of patterns are translated. Only 4862 entity pairs (5.05%) are translated (i.e., have parallel entity pairs). These very small ratios indicate that if we had relied only on machine translation, then we would not be able to achieve a reasonable recall level. We use SVDLIBC[5] to perform Singular Value Decomposition (SVD) of the matrix **A**. The matrix is a sparse matrix whose element density is only 0.01%. We set the value of k (the number of singular values to be calculated) to 300, as suggested by (Turney 2005).

In the monolingual clustering algorithm (without LRA), (Duc, Bollegala, and Ishizuka 2010) show that the appropriate value for θ_1 is 0.4. Therefore, we set the value of θ_1 to 0.4 in the **HSC** method (which uses $\mathrm{sim}_{\mathrm{VSM}}$) and vary the parameter θ_2 to determine the best value.

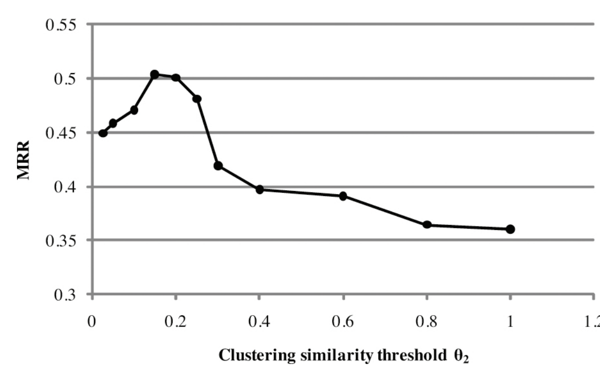

Figure 1: The relation between MRR and θ_2 of the **HSC** method (at $\theta_1 = 0.4$)

Fig. 1 shows the experiment result. At $\theta_2 = 0.15$, we obtain the best MRR. For two semantically similar lexical patterns p and q, $\mathrm{sim}_{\mathrm{LRA}}(p, q)$ is often larger than

[3]http://www.google.com

[4]Available at http://www.miv.t.u-tokyo.ac.jp/duc/milresh/

[5]http://tedlab.mit.edu/ dr/SVDLIBC/

$\mathrm{sim}_{\mathrm{VSM}}(p, q)$ because LRA compresses semantically similar dimensions into one and reduces noisy dimensions. Therefore, we can not assume that the appropriate value for θ_1 in the **HSC** method (which was set to 0.4) is also appropriate for the **HSC+LRA** method. However, we assume that the appropriate value for θ_2 (0.15) in the **HSC** method is also appropriate for the **HSC+LRA** method. This is because we need to associate as many parallel patterns as possible to the clusters to assure a high recall of the candidate retrieval process. Therefore, in all following experiments, we set θ_2 to 0.15.

We vary the value of θ_1 in the **HSC+LRA** method to find appropriate value. At $\theta_1 = 0.8$ we achieve the best performance for the **HSC+LRA** method. This value is much larger than the appropriate value in the **HSC** method (which was 0.4). Therefore, in all experiments related to the **HSC+LRA** method, we set θ_1 to 0.8.

Comparison with baseline methods

We compare the performance of the two proposed methods (**HSC** and **HSC+LRA**) with that of baseline methods using the test dataset (the 1.6GB corpus). Three baseline methods for comparison are as follows.

- **SC**: This method uses only the first phase sequential clustering in the Algorithm 1. This is the method in (Duc, Bollegala, and Ishizuka 2010) for monolingual latent relational search (however, **SC** finds parallel patterns by lexical pattern translation).
- **Trans+SC**: This method first translates all documents in the corpus into English. Then it translates all entities in the query into English and performs monolingual latent relational search, as in (Duc, Bollegala, and Ishizuka 2010).
- **LRA**: This method does not use clustering, instead it directly calculates the cosine similarity between two entity pairs using the dimensionally reduced vector space after LRA (i.e., the matrix $\mathbf{\Sigma}_k \mathbf{V}_k^T$).

The comparison is performed on eight query sets (four English-to-Japanese sets) similar to those in the previous section. The average MRR of eight query sets is shown in Fig. 2. The proposed methods outperform the **SC** method by

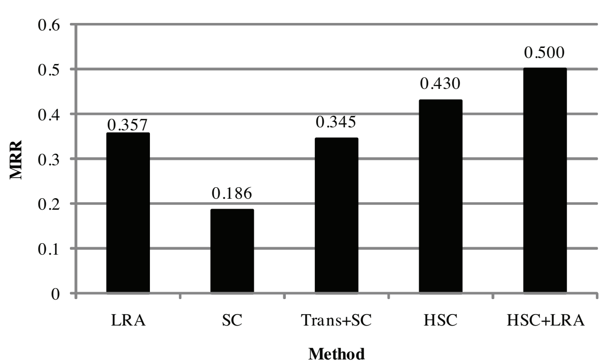

Figure 2: Comparison between the MRR of the proposed methods (**HSC** and **HSC+LRA**) with baseline methods

a wide margin. This proves that the proposed second-phase clustering successfully captures the semantic similarity between paraphrased lexical patterns across languages. Moreover, the two proposed methods outperform the **LRA** and **Trans+SC** method. The differences in the average MRR are statistically significant under the paired t-tests of 400 samples (from eight query sets, each containing 50 queries). Finally, the difference in MRR between the **HSC+LRA** method and **HSC** method is also statistically significant. This demonstrates that LRA significantly improves the performance of the system (with the cost of an SVD operation on a large matrix).

Performance of the system on several relation types

We use the test dataset (1.6GB corpus) to evaluate the performance of the system on 16 query sets (of eight relation types as describe in Table 1; eight of them are English-to-Japanese query sets, the rest are Japanese-to-English). The

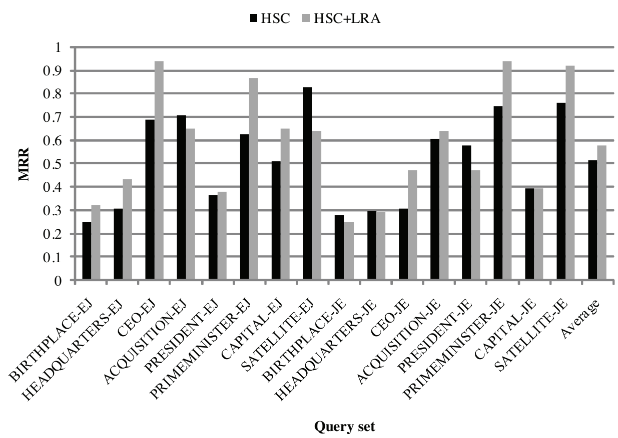

Figure 3: Performance of the two proposed methods for CLRS queries of eight relation types

evaluation result is shown in Fig. 3. The query sets subfixed with EJ are English-to-Japanese query sets. The CEO, SATELLITE and PRIMEMINISTER relation achieve good performance because the lexical patterns that represent these relations are easy to translate (in Japanese, sometime the phrase for describing the CEO relation is "CEO", which is identical with the phrase in English). On the other hand, the BIRTHPLACE and HEADQUARTERS relation have many different lexical patterns in Japanese and it is very difficult to exactly translate these patterns into English. Therefore, the performance on these query sets is not good.

Comparison with existing monolingual systems

We compare the performance of the proposed methods with that of two existing monolingual latent relational search systems (Kato et al. 2009; Duc, Bollegala, and Ishizuka 2010). We evaluate our system with eight Japanese-to-Japanese and eight English-to-English (monolingual) query sets corresponding to eight relation types in Table 1. The comparison result is shown in Table 2. The first row in the table shows the result reported in (Kato et al. 2009) on Japanese monolingual query sets (of many common relation types in

Table 2: Comparison between the proposed methods and existing methods (@N is the percentage of queries with correct answer in the Top N results)

Method	MRR	@1	@5	@10	@20
(Kato et al. 2009)[JJ]	0.545	43.3	68.3	72.3	76.0
(Duc, Bollegala, and Ishizuka 2010) [EE]	0.963	95.0	97.8	97.8	97.8
HSC-EE	0.967	94.1	99.8	99.8	99.9
HSC-JJ	0.888	87.5	90.0	90.0	90.0
HSC+LRA-EE	0.971	94.9	99.9	100	100
HSC+LRA-JJ	0.889	87.0	91.0	91.0	91.0
HSC-Cross	0.515	37.6	70.1	78.4	82.9
HSC+LRA-Cross	0.579	44.6	74.6	83.6	88.4

Table 1). The second row is the result reported in (Duc, Bollegala, and Ishizuka 2010) on English monolingual query sets (of four relation types in the first four rows in Table 1). The third and forth rows are the performances of the **HSC** method on English and Japanese monolingual query sets, respectively. The fifth and sixth rows are the performances of the **HSC+LRA** method on the same monolingual query sets. The two last rows are the performances of the proposed methods on cross-language query sets. Because we use the same extraction algorithm with (Duc, Bollegala, and Ishizuka 2010), the performance on monolingual query sets is at the same level with that of (Duc, Bollegala, and Ishizuka 2010). The performance of the **HSC+LRA** method on *cross-language* query sets is at the same level with that of the method in (Kato et al. 2009) on *monolingual* query sets. The gap between **HSC+LRA-EE** and **HSC+LRA-Cross** can be explained by the gap between the difficulty of monolingual latent relational search and cross-language latent relational search. The time for pre-processing a 1.6GB corpus is about one day. The time for processing a query of the proposed method is less than 10 seconds, which is acceptable for real-world search sessions.

Related work

There are many studies that address the problem of searching based on explicitly stated semantic relations (Tanaka-Ishii and Ishii 2007; Halskov and Barriere 2008; Banko et al. 2007) The method for latent relational search described in (Kato et al. 2009) represents the relations between two words in a given word pair by using the bag-of-words model. It does not require a local index for searching because it uses an existing keyword-based Web search engine to find the answer. However, the bag-of-words model does not allow the relational similarity between two word pairs to be precisely measured. To achieve a high precision, the relational similarity between (A, B) and (C, D) should be measured using a well-defined method such as (Turney 2005; Bollegala, Matsuo, and Ishizuka 2009), in which the relation between C and D is represented by lexical patterns that are in the same sentence with the pair (C, D). (Goto et al. 2010) exploit symmetries of semantic relations to improve performance of latent relational search engines.

Conclusion

We propose a hybrid sequential pattern clustering algorithm to capture the semantic similarity between paraphrased lexical patterns across languages. Using the result of this algorithm, we can precisely measure the relational similarity of two entity pairs in different languages and can therefore precisely rank the result list of a CLRS query. Moreover, we propose a method for combining LRA with sequential clustering to improve the performance of latent relational search. The system achieves an MRR of 0.579 on English/Japanese cross-language query sets while maintaining high performance on monolingual query sets. In future, we intend to apply the proposed method for corpora with more than two languages, such as corpora containing English, Japanese, Vietnamese and Sinhalese.

Acknowledgments

We would like to thank Prof. Peter D. Turney for his valuable suggestions and comments on this work. This research was partially supported by the IPA Mitoh Program of the Information-technology Promotion Agency of Japan and a grant from Google Research Award.

References

Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open Information Extraction from the Web. In *Proc. of IJCAI'07*, 2670–2676.

Bollegala, D.; Matsuo, Y.; and Ishizuka, M. 2009. Measuring the Similarity between Implicit Semantic Relations from the Web. In *Proc. of WWW'09*, 651–660.

Bollegala, D.; Matsuo, Y.; and Ishizuka, M. 2010. Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web. In *Proc. of WWW'10*, 151–160.

Bunescu, R., and Mooney, R. 2007. Learning to Extract Relations from the Web using Minimal Supervision. In *Proc. of ACL'07*.

Duc, N.; Bollegala, D.; and Ishizuka, M. 2010. Using Relational Similarity between Word Pairs for Latent Relational Search on the Web. In *Proc. of WI'10*, 196 – 199.

Goto, T.; ; Duc, N.; Bollegala, D.; and Ishizuka, M. 2010. Exploiting Symmetry in Relational Similarity for Ranking Relational Search Results. In *Proc. of PRICAI'10*, 595–600.

Halskov, J., and Barriere, C. 2008. Web-based Extraction of Semantic Relation Instances for Terminology Work. *Terminology* 14(1):20–44.

Kato, M.; Ohshima, H.; Oyama, S.; and Tanaka, K. 2009. Query by Analogical Example: Relational Search using Web Search Engine Indices. In *Proc. of CIKM'09*, 27–36.

Manning, C.; Raghavan, P.; and Schutze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

McNamee, P. 2005. Language Identification: a Solved Problem Suitable for Undergraduate Instruction. *Journal of Computing Sciences in Colleges* 20(3):94–101.

Tanaka-Ishii, K., and Ishii, Y. 2007. Multilingual Phrase-Based Concordance Generation in Real-Time. *Information Retrieval* 10(3):275–295.

Turney, P. 2005. Measuring Semantic Similarity by Latent Relational Analysis. In *Proc. of IJCAI'05*, 1136–1141.