A Framework for Cross-lingual/Node-wise Alignment of Lexical-Semantic Resources

Yoshihiko Hayashi
Faculty of Science and Engineering, Waseda University
2-4-12 Ohukubo, Shinjuku, Tokyo, 169-0072 Japan
yshk.hayashi@aoni.waseda.jp

Abstract

Given lexical-semantic resources in different languages, it is useful to establish cross-lingual correspondences, preferably with semantic relation labels, between the concept nodes in these resources. This paper presents a framework for enabling a cross-lingual/node-wise alignment of lexical-semantic resources, where cross-lingual correspondence candidates are first discovered and ranked, and then classified by a succeeding module. Indeed, we propose that a two-tier classifier configuration is feasible for the second module: the first classifier filters out possibly irrelevant correspondence candidates and the second classifier assigns a relatively fine-grained semantic relation label to each of the surviving candidates. The results of Japanese-to-English alignment experiments using EDR Electronic Dictionary and Princeton WordNet are described to exemplify the validity of the proposal.

Keywords: lexical-semantic resource, cross-lingual alignment, relation classification

1. Introduction

Given the current situation where a variety of language resources have been actively developed and disseminated, integrating these independently developed resources, preferably in a standardized format, has become an important issue (Cimiano et al., 2013). In particular, aligning lexical-semantic resources (henceforth, LSR) across languages in the word-sense/lexical-concept level is highly demanded for realizing cross-language semantic applications.

There are a few research projects in line with this research direction: BabelNet (Navigli and Ponzetto, 2012) maintains a wide-coverage multilingual semantic network, whose construction relies on mapping of Wikipedia (a multilingual encyclopedia system, in a sense) entries to the corresponding WordNet (Miller and Fellbaum, 2007) (an English semantic lexicon) senses; Uby (Gurevych et al., 2012) combines a variety of lexical resources for English and German also in a word-sense level by applying a machine-learned classifier.

Like these well-known projects, our primary research goal is to establish a computational method to discover cross-lingual correspondences in a word-sense/lexical-concept level, given LSRs in different languages. However we further explore a methodology for classifying discovered cross-lingual correspondences with a broader range of semantic relation types, not limited to synonymy. To pursue this research direction, we apply a machine-learned classifier as in Uby (Gurevych et al., 2012), but with a variety of semantic relation types, including near-synonymy, hypernymy, verb-argument, and so on.

This paper presents a whole system architecture for enabling *node-wise alignment* of LSRs in different languages, where potential cross-lingual correspondences are first discovered and ranked by applying an existing method (Hayashi, 2013), and then classified by a succeeding module. We particularly propose that a two-tiered classifiers configuration could be feasible for the second module: the first classifier filters out possibly irrelevant correspondences, and then the second classifier assigns a relatively fine-grained semantic relation label to each of the surviving correspondences.

To exemplify the proposal, we conducted a series of Japanese-to-English alignment experiments that employ EDR Electronic Dictionary (EDR; for Japanese)[1] (Yokoi, 1995) and Princeton WordNet (PWN; for English). EDR actually is a dictionary system, consisting of several types of computational dictionaries. and we mainly utilize EDR concept dictionary. Consult Appendix-A for more detailed description of EDR.

Here, we remind that we only utilize information given in Japanese for EDR, although this resource partly contains bilingual (Japanese and English) information. Similarly, we do not employ Japanese WordNet (Isahara et al., 2008), as our purpose is to explore a computational method to discover relevant cross-lingual correspondences from arbitrary LSRs.

2. Overall framework

2.1. The task: node-wise alignment of LSRs

Ontology matching, defined as a process for determining the alignment of a pair of given ontologies (Euzenat and Shvaiko, 2007), is one of the key technologies for enhancing the interoperability of existing knowledge resources, including LSRs. The importance of this technology has increased particularly in the context of Linked Open Data. Although, the ontology matching usually maintains structure-to-structure alignment, our work has preliminarily focused on the task of *node-wise alignment*, which could function as a fundamental building block for an entire structural alignment.

Figure 1 illustrates the node-wise alignment task, where one concept in the source LSR is chosen as a query (henceforth, a query concept), and the task is to discover a set of related concept nodes (including synonymous concepts) in the target LSR. Thus, as in ordinal information retrieval tasks, the problem is to achieve a higher precision while

[1] http://www2.nict.go.jp/out-promotion/techtransfer/EDR/index.html

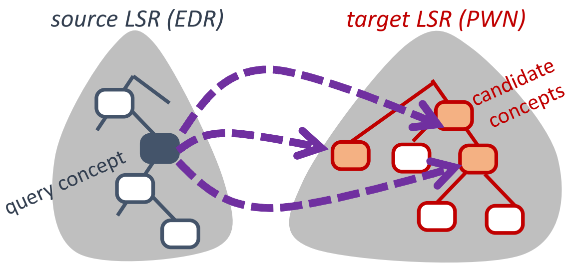

Figure 1: Node-wise alignment of concepts in different lexical-semantic resources. Dotted arrows represent potential semantic correspondences that should be classified and labeled.

keeping a level of recall. In addition, each correspondence (dotted arrows in Fig. 1) should be classified by the semantic type, and labeled accordingly. Among the possible types of semantic relation, *synonymy* would be of a particular importance especially in a cross-language setting, even though it could only dictate *near-synonyms* (Hirst, 2009).

2.2. Proposed architecture

Figure 2 displays our proposed architecture for enabling node-wise alignment of LSRs in different languages. First cross-lingual correspondence candidates are discovered and ranked by applying an existing method (Hayashi, 2013), and then filtered and classified by a succeeding module (green box in Fig. 2).

In particular, this paper discusses feasible configuration of the latter module. Although the filtering and the classification could be simultaneously performed by a single classifier, we propose that a two-tiered classifiers configuration could be more adequate. That is, the first classifier filters out presumably irrelevant correspondences, and then the second classifier assigns a relatively fine-grained relation label to each of the survived correspondence candidates.

3. Ranking candidate concepts

3.1. Cross-lingual semantic relatedness

A method to discover *conceptual mates* in the LSRs for other languages was proposed in (Hayashi, 2013), and this method could be instantly incorporated into our proposed architecture. The method, given a query concept s, produces a ranked list of candidate concepts (synsets) by computing $score(s,t)$ that measures the cross-lingual semantic relatedness between s and a candidate concept t. Here the set of candidate concepts is first generated by exploiting bilingual translation resources (Hayashi, 2012a).

$$score(s,t) \equiv (1-\beta)pscore(s,t) + \beta gscore(s,t) \quad (1)$$

As defined in the formula, $score(s,t)$ is computed as the weighted sum of $pscore(s,t)$ (*synonym-based relatedness*) and $gscore(s,t)$ (*gloss-based relatedness*), where $0.0 \leq \beta \leq 1.0$ balances the blending of these elements.

The synonym-based relatedness $pscore(s,t)$ dictates a cross-lingual semantic relatedness based on synonymous words (i.e., words that jointly specify a lexical concept). This score is computed by employing bilingual resources along with a sense-tagged corpus in the target language (Hayashi, 2012a).

The gloss-based relatedness $gscore(s,t)$, on the other hand, measures a cross-lingual semantic relatedness based on the textual similarity between the (extended) glosses (Banerjee and Pedersen, 2003). As the languages of comparing gloss texts are different, the extended gloss (in Japanese) for the query concept is first machine-translated[2] into the target language (English), and then a monolingual textual similarity measure is applied.

The described configuration can be considered reasonable, as textual glosses, in general, provide richer linguistic contexts and could be better machine-translated compared to the set of synonym words given to a concept node.

3.2. Ranked list of candidate concepts

The results reported in (Hayashi, 2013) can be summarized as follows: (1) Synonymous concepts were ranked first for almost half (47.4%) of the test query concepts; (2) Somewhat related concepts (including other than synonymous concepts) were discovered with a relatively good performance: mean average precision (MAP) was 0.695; (3) The optimum value of β was around 0.6, but rather robust in the range of [0.4, 0.8].

Table 1 shows a part of the ranked list for an EDR query concept (roughly means *big power*), exemplifying that a ranked list is a mixture of synonymous concepts (flag=*syn*), some related concepts (flag=*rel*), as well as irrelevant concepts (flag=*irrel*). This exactly motivates the presented work. That is, we would like to filter out irrelevant correspondences, and assign a label that signals the type of semantic relation to each of the relevant correspondences.

Notice also that the parts-of-speech of candidate concepts are not uniform in the ranked list. This is brought about by the nature of EDR: an EDR concept does not explicitly load part-of-speech information, exhibiting a yet another notable difference between PWN and EDR.

4. Classifying semantic relations

4.1. Overall approach

We adopt a supervised learning approach. That is, we train a classifier, given hand-annotated training data. To do this, first we choose a set of query concepts from the source LSR (in this work, EDR), and then collect the ranked lists of candidate concepts in the target LSR (PWN): the method described in the previous section is applied for this purpose. Each cross-lingual correspondence candidate in the ranked lists is next hand-classified as one of the semantic relation types defined in an inventory (section 4.2). The annotated ranked lists are then fed into a machine learning (ML) algorithm, which employs a set of presumably effective features (section 4.3). The resulting classifier is finally evaluated by employing standard evaluation measures (section 4.4).

4.2. Annotating cross-lingual correspondences

Table 2 summarizes our inventory of semantic relation types, which is referred in the creation of training data. Although we initially designed a more fine-grained inventory

[2] We utilized Web-based translation services provided by the Language Grid (`http://langrid.org/`).

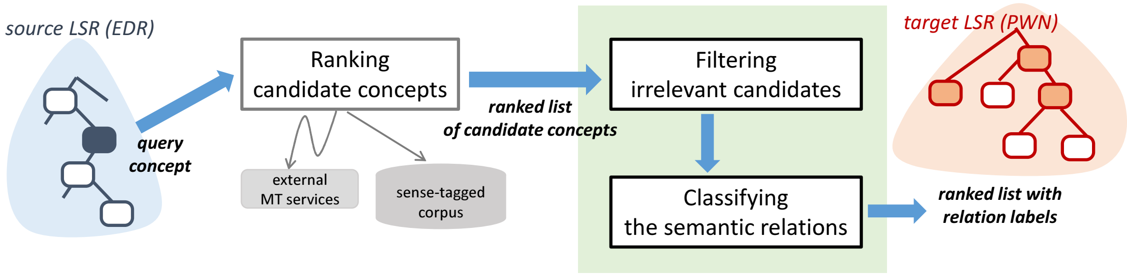

Figure 2: Proposed architecture.

r	Synset ID	Synonymous words	Gloss	flag
1	13945102-n	power; office	holding an office means being in power	*rel*
2	11453016-n	power	the rate of doing work;measured in watts (= joules/second)	*irrel*
3	05190804-n	power; powerfulness	possession of controlling influence	*rel*
4	05616246-n	ability; power	possession of the qualities ... required to do something or get something done	*rel*
5	08177592-n	superpower;great power; major power; world power	a state powerful enough to influence events throughout the world	*syn*
6	01181559-v	power	supply the force or power for the functioning of	*irrel*
7	05030680-n	mightiness; might; power	physical strength	*irrel*
8	10461424-n	power; force	one possessing or exercising power or influence or authority	*rel*

Table 1: A part of the ranked list of candidate PWN synsets for the query EDR concept with ID:0fb994 (represented by a Japanese word ”大国”, also translated as ”world power; powerful country”).

Type	Explanation
synonym	s and t share an almost identical meaning
near-synonym	s and t share a similar meaning
broader-term	t denotes a broader meaning than s
narrower-term	t denotes a narrower meaning than s
argument	t can be one of the arguments of s
causality	s can cause a situation denoted by t
stative	t denotes a stative derivation of t
miscellaneous	s and t somehow related (including antonyms)
irrelevant	t has nothing to do with s

Table 2: Inventory of semantic relation types.

by referring to the semantic relation types developed by the EuroWordNet project (Vossen, 1998), we have ended up with the current one, where modest simplification was applied[3]. Notice that *broader-term* includes hypernyms and holonyms, while *narrower-term* includes hyponyms and meronyms/partnyms. Also notice that *irrelevant* is introduced as a special semantic relation types.

4.3. Features for machine learning

Features for representing a correspondence candidate can be divided into two feature groups: (1) features for a source/target concept node, and (2) features for a source-target node pair. We create a feature vector for each of the correspondence candidates by concatenating the sub-vectors for representing these feature groups.
Feature groups for a concept node are:

- Upper semantic classes: In order to signify the coarse semantic nature of a concept, we assign the set of upper semantic classes as a feature of the concept. For this purpose, the lexicographer files, 45 in total, were adopted as the upper classes for PWN; 36 concepts, placed relatively higher in the concept hierarchy, were selected as the upper classes for EDR. We can easily assign this type of semantic feature even for an EDR concept by navigating the concept hierarchy. This type of semantic feature is finally represented by a 45 (for PWN) or a 36 (for EDR) dimensional binary vector.

- Part-of-speech: The part-of-speech of a concept is also employed as a feature and represented by a 5-dimensional binary vector. PWN explicitly provides this information, but EDR does not. To overcome this issue, we have assigned a possible set of parts-of-speech to each of the EDR upper semantic concepts, and we fake the POS information of an EDR query concept by referring to this assignment.

- Graph features: Two types of features computed from the graph structure of an LSR are introduced to dictate the relative importance of a concept node.
  - Depth: hop count of the node from the root node of the LSR.
  - Node centralities (Mihalcea and Radev, 2011):

[3]Our annotators were largely puzzled with the initial inventory. In particular, the distinction between hyponymy with meronymy was not only hard for them but sometimes impossible to do, suggesting that the task of cross-lingual classification of semantic/conceptual relations for independently developed LSRs involves innate difficulties.

betweenness centrality and load centrality provided by NetworkX[4] Python module.

Features for a source-target node pair are further divided into two types. It is natural to incorporate the scores given by the method described in section 2.1. They are: $score(s,t)$ in the left hand side of the eauation (1), and $pscore(s,t)$ and $gscore(s,t)$ in the right hand side. In addition to these scores, we tacked two types of gloss similarity as described below.

Word embedding vector-based gloss similarity: We have assigned a semantic vector for each of the gloss text, and computed the cosine similarity. Remind that the gloss text in the source language is translated into the target language: in this case Japanese into English. The semantic vector for a gloss text (both in English) is constructed by summing up[5] the word embedding vectors for each words in the gloss text. Each word vector is learned from a relatively large text corpus[6] by applying word2vec method (Mikolov et al., 2013).

Alignment-based gloss similarity: Another gloss similarity relies on the recently proposed monolingual sentence alignment algorithm (Sultan et al., 2104), which achieves a good level of accuracy due to the exploitation of dependency syntax. We incorporate the word embedding-based word similarity (with the threshold 0.65), instead of the prefixed synonym dictionary utilized in the original version, to admit word-level correspondences. Note that we utilize multiple MT engines to translate a source gloss text, yielding a set of translated gloss texts. Therefore the maximum similarity achieved from one of the translated gloss texts and the target gloss text is adopted as this alignment-based gloss similarity.

4.4. ML algorithm and the evaluation measures

We train a classifier by adopting the Random Forest® (Breiman, 2001) algorithm provided by scikit-learn[7] Python module. The Random Forest algorithm was utilized, since it constantly gave us better results compared to the Support Vector Machines in the preliminary experiments. As our problem is a typical multi-class classification, we evaluate the learned classifier with the standard performance measures: accuracy, precision, recall, and F1.

5. Experiments

5.1. Objectives

The objectives of the experiments are three-fold:

- To decide which classifier configuration is better: a one-tier classifier configuration where a single classifier also filters out irrelevant candidates, or the two-tiered classifiers configuration as shown in Fig. 2;
- To compare among alternative systems of semantic relation types: A system that results in a better classification performance (with respect to some perspective) should be explored;
- To investigate into the importance of a proposed feature group.

To accomplish these objectives, we collected the performance measures on the ranked lists data (section 5.2), while altering the system of semantic relation types (section 5.3). With respect to the last objective, we have conducted so-called ablation tests, which will be detailed afterwards (section 6.3).

Type	# of counts (%)
synonym	2,406 (11.4)
near-synonym	1,546 (7.3)
broader-term	1,242 (5.9)
narrower-term	1,094 (5.2)
argument	597 (2.8)
causality	340 (1.6)
stative	302 (1.4)
miscellaneous	119 (0.6)
irrelevant	13,393 (63.6)

Table 3: Distribution of semantic relation types.

5.2. Data

5.2.1. Query concepts and the ranked lists

First we collected a set of ranked lists of at most 10 candidate PWN synsets generated for an EDR query concept. To run the method described in (Hayashi, 2013), we selected 2,300 EDR query concepts by roughly observing the distribution of concepts in terms of a higher-level semantic classification.

Among the 2,300 query concepts, the method yielded effective ranked list for 2,036 queries (effective rate: 87.2%). Most of the ineffective queries were associated with some domestic concepts and/or named entities.

Calculated from the annotation results (described in the next subsection), the MAP measures for synonymous and related concepts are 0.483 and 0.783 respectively. These are slightly better figures than the ones reported in the original literature.

5.2.2. Distribution of semantic types

A team of annotators went through the ranked lists, and assigned one of the semantic types chosen from the inventory shown in Table 2. Although each of the ranked lists was checked by only one annotator, the final results were approved through discussions by the annotator team.

Table 3 counts the distribution of the semantic relation types given to the data, suggesting that filtering of irrelevant candidates is necessary, and further showing that the distribution is skewed even without considering irrelevant ones. Remind that the large portion of irrelevant candidates does not necessarily mean that the employed method performed only poorly, because we retained at most 10 candidate synsets for each query concept.

[4] `https://networkx.github.io/`

[5] We have tried a few variety of mathematical operations but the simple addition gave the best results.

[6] enwik9 corpus is available for download at: `http://mattmahoney.net/dc/text.html`

[7] `http://scikit-learn.org/`

Mnemonic of the system	Description
`binary`	irrel: {*irrelevant*}, rel: (other relation types)
`fine`	syn: {*synonym*}, near-syn: {*near-synonym*}, brd: {*broader-term*}, nrw: {*narrower-term*}, irrel: {*irrelevant*}, misc: (other relation types)
`middle`	syn: {*synonym*}, rel: {*near-synonym*, *broader-terms*, *narrower-terms*}, irrel: {*irrelevant*}, misc: (other relation types)
`coarse`	syn: {*synonym*}, rel+: *rel* ∪ *misc*

Table 4: Alternatives of relation type system.

Mnemonic	N	A	P	R	$F1$
`binary`	2	**0.740**	0.7566	0.8647	0.807
`fine`	6	0.655	0.678	0.972	0.799
`middle`	4	0.667	0.702	0.945	0.806
`coarse`	3	0.675	0.719	0.927	0.810

Table 5: Classification performances for *irrelevant* relation. (N represents the number of classes (types); Accuracy applies to all target types.)

5.3. Possible systems of semantic relation types

Although the final applicability of a system of semantic relation types depends on its final application, the distribution (Table 3) suggests some modification to the inventory (Table 2) should/could be made. To empirically investigate this issue, we compare four semantic relation type systems shown in Table 4. Notice that the `binary` system is particularly introduced to assess the two system-architectural alternatives discussed in section 5.1.

6. Results

The results presented in this section were collected from a series of experiments, in which the Random Forest learning algorithm was adopted.

We applied a standard 5-fold cross validation procedure: therefore the reported figures were computed by averaging over the results of the five trials. Also note that we confirmed the variations in the results, brought about by the nature of the Random Forest, were very small.

6.1. Filtering irrelevant correspondences

Table 5 summarizes the classification performances, particularly focusing on the detection of irrelevant correspondences. The important remark here is that the shown results were obtained by the single classifier configuration, where *irrelevant* is included in the set of target classes.

Although the best $F1$ score was achieved with the `coarse` type system, the `binary` type system could be considered most adequate, because it moderately balances the precision (P) and the recall (R), and yields the best accuracy

Mnemonic	N	A	P	R	$F1$
`fine`	5	0.438	0.426	0.674	**0.522**
`middle`	3	0.651	0.539	0.292	0.378
`coarse`	2	**0.712**	0.572	0.219	0.316

Table 6: Classification performances for *synonym* relation.

Type	P	R	$F1$
synonym	**0.426**	**0.674**	**0.522**
near-synonym	*0.413*	*0.388*	*0.400*
broader-term	0.341	0.223	0.270
narrower-term	0.271	0.111	0.158
miscellaneous	0.747	0.608	0.670

Table 7: Classification performances of the `fine` system (without irrelevant correspondences).

(A) of 0.740, which insists that this configuration also gave a better performance for relevant types.

Therefore, we would conclude that irrelevant correspondences should be first filtered out, which means we should adopt the two-tiered classifiers configuration that was displayed in Fig. 2.

6.2. Comparison of semantic relation type systems

Although the ”binary filter” works at a reasonable performance level ($F1$ is around 0.8 with balanced P and R), it is still far from perfect. Nevertheless, here we assume that (a future version of) the filter perfectly performs, and compare the alternatives of the semantic relation type systems classified in Table 4, particularly in terms of the classification performances of synonymous correspondences.

Table 6 compares the classification performances of *synonym* relation, where only the non-irrelevant data are fed into the machine learning process. Notice that the `binary` system disappears here by definition, and the number of classes is decreased by one in all systems.

It is natural to observe the A increases as the number of target types (classes) decreases. Interestingly, however, the $F1$ measure for *synonym* relation is rather better with the `fine` system ($F1 = 0.522$) than that with other semantic type systems. This may due to: if we collapse *dissimilar* classes into a single class or a few classes, the total classification performance might degrade, suggesting that we have to adopt an appropriate semantic relation type system.

Table 7 further breaks down the classification performances of the `fine` system for each classes, showing that the results for *synonym* and *near-synonym* are somewhat promising, but *broader-term* and *narrower-term* are hopelessly difficult to discriminate. This may partly reflect the distribution of the semantic relation types in the data shown in Table 3. Nevertheless, these difficult classes may indirectly contribute to the classification of *synonym* relation, as well as that of *near-synonym* relation probably.

In summary, the results presented so far would justify our proposal on the system-architectural issue. That is, the two-tiered classifiers configuration could be more feasible for classifying cross-lingual correspondence candidates be-

Mnemonic	A	P	R	$F1$
full	0.438	0.426	0.674	0.522
-scores	**0.412**	**0.3939**	**0.626**	**0.483**
-sem	0.428	0.422	0.655	0.514
-align-sim	0.436	0.422	0.657	0.513
-w2v-sim	0.437	0.421	0.674	0.518

Table 8: Ablation test results (without *irrelevant* correspondences).

Mnemonic	A	P	R	$F1$
full	0.655	0.459	0.309	0.370
-scores	**0.639**	**0.390**	**0.221**	**0.282**
-sem	0.654	*0.436*	0.319	0.369
-align-sim	*0.653*	0.458	0.277	0.345
-w2v-sim	0.650	0.467	*0.272*	*0.344*

Table 9: Ablation test results (with *irrelevant* correspondences).

tween different LSRs. However further experimentations with a larger data set would be necessary to make this more concrete.

6.3. Importance of a feature group

We conducted a series of ablation tests in order to explicitly investigate the importance of each feature group. Table 8 displays the results with the `fine` system on the data without irrelevant candidates, where each mnemonic dictates one of the following ablation conditions. Larger degrade in the performance measure means that the associated feature group was effective hence important.

- full: without any ablations (for comparison).
- -score: without scores in the candidate ranking ($score$, $pscore$, and $gscore$).
- -sem: without the upper semantic classes feature.
- -align-sim: without the alignment-based gloss similarity.
- -w2v-sim: without word embedding vector-based gloss similarity.

As expected, the score features play the most prominent role, being consistent with the results of the candidate ranking module, where MRR for the *synonym* relation was relatively high as around 0.5. Each of the remaining three feature groups also plays a role, but it should be noted that the word2vec-based gloss similarity was not very effective, probably suggesting that it captures more vague semantic relatednesses rather than semantic similarities.

For a comparison purpose, Table 9 shows corresponding performance figures obtained from the data including irrelevant correspondences. Like the previously shown results, the score feature group contributed most. Unlike the previous results however, the word2vec-based gloss similarity was confirmed to be effective in the classification, showing that semantic relatedness captured by this feature could contribute to distinguishing irrelevant correspondences from relevant ones.

7. Concluding remarks

This paper empirically showed that a two-tiered classifiers configuration is feasible for classifying cross-lingual correspondences between concept nodes in different lexical-semantic resources. The resulting performance measures revealed that the task itself is quite challenging, presumably due to innate differences in the *semantic ranges* of concepts in the languages. Often the information explicitly or implicitly provided by each concept in a lexical-semantic resource are not suffice to distinguish between them from competing candidate concept in the other resources. This means that we may need to explore a better inventory of semantic relation type system as well.

Technically, we may have more rooms to improve. We would be able to incorporate more effective constraints/preferences from the structure of each LSRs. Besides, recent progresses in distributed/distributional semantic representations (Weeds et al., 2014; Bollegala et al., 2015; Rothe and Shütze, 2015) could help improve the set of features utilized in the machine learning process.

8. Acknowledgments

This work was supported by JSPS KAKENHI Grant Number #25280117.

9. Bibliographical References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pp.805–810.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015 Embedding semantic relations into word representations. *Proc. of the 24th International Joint Conferences on Artificial Intelligence (IJCAI 2015)*, pp.1222–1228.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45 (1): 5–32.

Phillip Cimiano, John McCrae, Paul Buitelaar, and Elena Montiel-Ponsoda. 2013. On the role of senses in the ontology-lexicon. In: Alessandro Oltamari et al. (Eds.), *New Trends of Research in Ontologies and Lexical Resources.*, pp.43–62. Springer.

Jérôme Euzenat and Pavel Shvaiko. 2007. *Ontology Matching*, Springer.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmannm, Michael Matuschek, Christian M. Meyer, and Christian Wirth. 2012. UBY - a large-scale unified lexical-semantic resource based on LMF. *Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pp.580–590.

Yoshihiko Hayashi. 2012a. Computing cross-lingual synonym set similarity by using Princeton annotated gloss corpus. *Proc. of 6th International Global WordNet Conference (GWA2012)*, pp.131–141.

Yoshihiko Hayashi, Savas Ali Bora, Monica Monachini, Claudia Soria, Nicoletta Calzolari. 2012b. LMF-aware

Web services for accessing semantic lexicons. *Language Resources and Evaluation*, *46*(2), pp.253–364.

Yoshihiko Hayashi. 2013. Blending two kinds of semantic relatedness for cross-language matching of lexical concepts. *Proc. of 10th International Conference on Terminology and Artificial Intelligence (TIA2013)*, pp.35–42.

Graeme Hirst. 2009. Ontology and the lexicon. In Steffen Staab and Rudi Studer (Eds.) *Handbook of Ontologies*, pp.269–292. Springer.

Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama and Kyoko Kanzaki. 2008. Development of Japanese WordNet. *Proc. of the Sixth International Conference on Language Resources and Evaluation (LREC2008)*.

Rada Mihalcea and Dragomir Radev. 2011. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*, pp.3111–3119.

George Miller, and Christiane Fellbaum. 2007. Wordnet. then and now. *Language Resources and Evaluation 41*, 209–214.

Roberto Navigli, and Simone Paolo Ponzettoo. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Sascha Rothe and Hinrich Shütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp.1793–1803.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, Vol.2, p.219–230.

Piek Vossen (Eds.) 1998. *EuroWordNet*. Kluwer Academic Publishers.

Julie Weeds, Daud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. *Proc. of the 25th International Conference on Computational Linguistics (COLING 2014)*, pp.2249–2259.

Toshio Yokoi. 1995. The EDR electronic dictionary. *Communications of the ACM 38*(11), 42–44.

Appendix-A. Information structure of the EDR Electronic Dictionary

The EDR electronic dictionary, unlike Princeton WordNet, is not a lexical database based on relational lexical semantics: Rather, it can be seen as a knowledge base enriched by linguistic information given in both Japanese and English. Another big difference lies in the consideration of parts of speech in the conceptual organization: as well known, PWN maintains POS-dependent lexical semantic networks, while EDR bears only a POS-independent network. Despite these differences, the information structure of EDR can be modeled in the same way as that of PWN (Hayashi, 2012b): the set of words associated with a common concept identifier in one or more of the sub-dictionaries can be modeled as a kind of synset. In addition, EDR defines a set of types of conceptual relations linking one concept to another, which are structurally the same as in PWN.