Active Learning-based Local Graph Matching for Textual Entailment

Tsuyoshi Okita
Dublin City University, School of Computing
tokita@computing.dcu.ie

Abstract

This paper presents a robust textual entailment system using the principle of *just noticeable difference* in psychology, which we call a local graph matching-based system with active learning. First, although an early textual entailment task often involved two rather simple sentences of T ("Text") and H ("Hypothesis"), the recent textual entailment task often involves multiple / complex / compound sentences in T (and H). In this context, the search for corresponding subgraphs of T and H is one important task with regard to the principle of *just noticeable difference* where irrelevant items will work as noise for the similarity-based graph-matching approach. Our approach concerns to reduce such noise by linguistic preprocessing being conscious of the topic mentioned by H via subject alignment. Second, by definition of textual entailment task, one interesting task involves the understanding of T which will not be detected by the similarity-based method. Third, one bottleneck of textual entailment task is the performance drop by unknown words and named-entities. We explore reducing the unknown words and named-entities, incorporating meaning in parentheses / rhetorical expressions / semantic roles, and additional feature of language model from deep learning. Our result for RTE2 corpus was 80.49 for macro F1 score, 84.95 for precision for the positive entailment, and 79.95 for recall for negative entailment.

Keywords: Textual Entailment, Semantics.

1 Introduction

A textual entailment task addresses the variability of semantic expression whether the same meaning can be expressed by or inferred from different texts [1]. A graph matching approach [2] is one classical approach towards textual entailment. Sentences are represented as normalized syntactic dependency graphs and entailment is approximated with an alignment between the graph representing the hypothesis and a portion of the corresponding graph(s) representing the text. Although this approach is attractive, we require at least two extensions.

First, the basic limitation of this approach is that the graph matching is done using global features where other material in T (or H) may affect the results of the match. This includes a small example: dropping of a restrictive modifier does not preserve entailment in a negative context since the search is based on global features. All the more, there is one important change in its setting: although early textual entailment task often involved two simple sentences of T and H, the recent textual entailment task often involves multiple / complex /compound sentences in T (and H). In this context, the search for corresponding subgraphs of T and H becomes one important task where irrelevant items will work as noise for the similarity-based graph-matching approach. MacCartney et al. [3] employ typed dependency graphs, partial alignment between the typed dependency graphs representing the hypothesis and the text, and a decision of entailment. They mentioned the confounding between the alignment and decision processes. Mirkin et al. [4] deploy a slightly different approach where they use a partial graph which they call a subsentential textual entailment. This paper refers the principle of *just noticeable difference* in psychology / color perception [5] for guiding the search. This principle says the danger of comparing two quantities which have a big difference. Hence, we tried to decide whether this T entails H or not within a single aspect with very small difference by reducing the noise by linguistic preprocessing being conscious of the topic mentioned by H via subject alignment. Among various ways to confirm this principle, the way we perform in this paper is to avoid the comparison of two sentences when the distance involves mixture of lexical distance with different grammatical functions, grammatical distance with different subject, distance between unknown words, and distance which requires the understanding of the text which is beyond the capability of the similarity-based comparison. We tried to first preprocess them into a graph which can be manageable within the framework of lexical distance. One important preprocessing among them is the subject alignment. Our method is slightly different from the alignment approaches by MacCartney et al. [3] or Bentivogli et al. [6] which align all the possible elements in T and H. Bentivogli et al. [6] further handles the coreferences in the context of discourse. These approaches have a problem of confounding of alignment and decision processes, as is mentioned by MacCartney et al. [3]. Our approach detects the corresponding subjects which sometimes requires to detect voices or semantic roles, which will facilitate the decision phase in the same reason. Furthermore, our system can also handle the resolution of time / space references are in itself necessary to determine whether T entails H.

Second, by definition of the textual entailment task, we believe that one interesting task for modern textual entailment task would involve the understanding of T. A simple example of this is that the number of person is not explicitly mentioned but simply the names of person in T, while the number of person is explicitly mentioned in H. This task is potentially difficult if we use the similarity-based matching. If a testset includes such sentences which require the understanding of T, the ability to understand text will improve the overall score. This paper shows only a simple solution in a very limited manner which can only solve the simple case. This is not because our approach can extend without loss of generality but the approach which will solve such problem in a general manner is really hard for us to provide in this paper.

Finally, we note that this paper uses active learning. Active learning often discovers new training points other than the provided training points. We actively discover new sub-points adding on the substructures of T and H in order to find closer substructures T' and H'. We actively discover indispensable additional training sub-points by linguistic preprocessing: we supply the unknown words and unknown named-entities, give meaning in parenthesis / rhetorical expressions / semantic roles, and prepare additional information in order that we can perform (simple) text understanding.

The remainder of this paper is organized as follows. Section 2 describes the review of graph matching model. In Section 3, we describe the principle of just noticeable difference and we present our model in Section 4. Our experimental results are presented in Section 5. We conclude in Section 6.

2 Review of Graph Matching Model

As is similar with Haghighi et al. [2], we represent text of T and H as a graph in the following way. First, T and H are represented as a dependency tree using the modified version of Collins' head propagation rules, i.e. main verbs are placed at the head of sentences. Second, the dependency nodes such as collocation and named-entities are collapsed. Note that collocation include verbs and their adjacent particles. Third, certain dependencies such as modifying prepositions are folded. Fourth, the graph representation is augmented by Propbank-style semantic roles. Each predicate adds an arc labeled with the appropriate semantic role to the head of the argument phrase. Modifying phrases are labeled with their semantic types.

The summary of the graph matching model introduced by Haghighi et al. [2] is as follows. Let H denote hypothesis graph, T denote a text graph, M denote a mapping from the vertices of H to those of T, $M(v)$ denote the match in T for vertex v in H , and Cost(M) be the cost of matching M. When M is a set of matching, the cost of matching H to T is defined as in (1):

$$\text{MatchCost(H,T)} \;=\; \min_{M \in \mathcal{M}} \text{Cost}(M) \quad (1)$$

where Cost(M) is given by a convex mixture of the node and relational match costs as in (2):

$$\text{Cost(M)} = \alpha \text{Cost}_N(M) + (1-\alpha)\text{Cost}_R(M) \quad (2)$$

where Cost$_N(M)$ denotes a node cost, and Cost$_R(M)$ denotes relational match cost. Let Sub$_N$(v,M(v)) be a model for substituting node v for $M(v)$. Then, the node cost is represented as in (3):

$$\text{Cost}_N(M) = \frac{1}{Z}\sum_{v \in H_v} w(v)\text{Sub}_N(v, M(v)) \quad (3)$$

where $w(v)$ denotes the weight for node v, and Z (=$\sum_{v \in H_v} w(v)$) denotes a normalization constant. Similarly, let Sub$_P$(e,ϕ_M(e)) be a model for assessing the cost of substituting a direct relation e for $\phi_M(e)$ under the matching. Relation cost is represented as in (4):

$$\text{Cost}_R(M) = \frac{1}{Z} \sum_{v \in H_v} w(e)\text{Sub}_P(e, \phi_M(e)) \quad (4)$$

where $w(e)$ denotes a edge cost, and Z (=$\sum_{v \in H_v} w(e)$) denotes a normalization constant. In sum, this model yields T entails H when MatchCost(H, T) is low, and otherwise T does not entail H.

3 Danger to Compare Big Difference (Just Noticeable Difference)

This paper considers the meaning of difference in T and H which are described in natural language. The semantic distance between two words has various established theories such as distributionally-based semantic similarity [7], taxonomy-based semantic similarity [8] and information-based similarity [9]. However, the semantic distance between two sentences has still difficulty in various places until establishing a synthetic framework. This is since their difference will involve various combinations of lexical difference, grammatical difference, and their difference of information content, while the overall semantic difference of two sentences should be measured by a single measure.

For instance, we suppose that we compare H and T below. there would be several possibilities where the parenthesis () denotes the comparison and the first element comes from H and the second element from T: (1) we compare (Bob, Mary), (bought, sold), (a red car, a car), (Mary, Bob), and (2) we compare (Bob, Bob), (bought, sold), (a red car, a car), (Mary, Mary). (1) mixes various things and (2) has some difficulties how to quantify (bought, sold).

H:	Bob bought a red car from Mary.
T:	Mary sold a car to Bob.
T':	Bob bought a car from Mary.

Our method derives intermediate representation T' by preprocessing and compare H and T'. The validity of this approach comes from an analogy of the method in color perception [5] or other psychology, in which we avoid to compare a big difference.

The color perception may compare the two big distances if two colors are distant. First, the just noticeable difference is the smallest detectable difference of a particular sensory stimulus, which is due to Ernst Heinrich Weber who studied the human response to a physical stimulus in a quantitative fashion in 19th century. Second, the trichromatic color theory in color perception was the results of Thomas Young in 18th century, which showed that normal vision needed three wavelengths to create the range of colors. (Now we know that we have three types of color-sensing cone cells. In this case, the just noticeable difference corresponds to each basis of wavelengths in red, green, and blue, respectively.) Under such circumstances, if we quantify the combined effects of sensory stimulus whose effects are linear, we can separate each effects. However, the combined effects of color perception are known to be nonlinear. If the combined effects are nonlinear, it is often very difficult and dangerous to quantify the distance between two colors if they are distant. Usually, it is better to avoid the comparison of two quantities whose difference is big. Nevertheless, it is safe to compare two quantities if they are within a small difference.

Now we go back to our situation of comparison of two sentences. The graph-based matching approach compares two sentences with a single measure. This becomes problematic since the comparison involves combined effects such as lexical difference, grammatical difference and the difference of information contents. Hence, our approach limits ourselves within a small difference: to search subgraphs T' and H' which have preferably a single (and small) difference.

It is noted that this approach may be explained in terms of matrix factorization. Equation (2) says that Cost(M) is the convex combination of Cost$_N(M)$ and Cost$_R(M)$. The linguistic preprocessing, described in Section 4, intends to reduce the dimensionality (or the rank) of matrices Cost$_N(M)$ and Cost$_R(M)$ if we see these as matrices where we especially intend to reduce the relation cost Cost$_R(M)$. The more the dimensionality (or the rank) of Cost$_N(M)$ and Cost$_R(M)$ reduced, the more we can avoid to compare a big difference.

4 Our Model

Now based on the graph matching algorithm, we convert this into mathematical expressions by introducing the locality. Let T_j be subgraph of T and H_i be subgraph of H. For example, when T consists of multiples of sentences, T_j may be a simple sentence. Let $H_i \approx T_j$ denote that H_i is close enough to T_j. We suppose that the comparison of the irrelevant elements of H_i and T_j will be zero. For instance, T: John and Mary is 5 years old. H: John is 5 years old. The information about Mary is irrelevant for the sake of H. Hence, we convert T into T_2: Mary is 5 years old, $\text{Cost}_l(T_2, H, M_{(2,1)}) = 0$. [1]

In order to avoid the assessment of the cost globally, we decompose $\text{Cost}(H, T, M_{(i,j)})$ with a set of $\text{Cost}_l(H_i, T_j, M_{(i,j)})$ where a decomposed subgraph includes a close pair of H_i and T_j which satisfies $H_i \approx T_j$. Hence, the modified version of the MatchCost(H,T) can be written as in below:

$$\begin{aligned} \text{MatchCost(H,T)} &= \min_{M \in \mathcal{M}} \text{Cost}(H, T, M_{(i,j)}) \\ \text{Cost}(H, T, M_{(i,j)}) &= \sum_{H_i \approx T_j} \text{Cost}_l(H_i, T_j, M_{(i,j)}) \end{aligned}$$

Note that this $H_i \approx T_j$ corresponds to the principle of the just noticeable difference in psychology. Although this indicates that for given T and H it may not be possible to find out such (a set of) H_i and a set of T_j. Other note is that if H_i only refers a subset of T, it may not need to consider other part of T. This means that we may not need to iterate all subsets of T to compare H_i. This also means that we treat this as if a set of H_i and T_i is almost mutual exclusive and only a couple of pairs of H_i and T_i is active in practice.

4.1 Deep / Shallow Linguistic Preprocessing Step

4.1.1 Subject Alignment on Dependency Structure

The deep / shallow linguistic preprocessing step modifies the original structures of H and T in order to provide H_i and T_j which are subsets of H and T with corresponding features in the classification step. Using the equation (5) and (6), our algorithm makes the size of the source and the target sentences shrinked in order that we can compare the H_i and T_j where H_i and T_j are close enough and where other local cost $\text{Cost}_l(H_i, T_j, M_{(i,j)})$ can be considered to be infinity (although this is not always the case; all the more, only the pair of H_i and T_j will make matching while other pair will not). In this process, the structure of texts are actively investigated in two directions: (1) make T from complex / compound sentences into simple sentences and (2) make the form of T simplified with considering the easier match with H. At the same time, the feature extraction are actively proceeded in order to help the simplification of T. We call this mechanism as *active learning* since the features used in the standard SVM are not modified but are globally evaluated. Note that although it is often the case that active learning let increase the training data, the active learning here let decrease the substructure of training data and let extract the features according to this dynamical substructure.

The conversion of subject relations (or subject alignment) of T_i towards H_i is the second topic in this process. It aims at aligning the subject in H to the phrase T_s which is a subgraph of T. If T_s is not the subject in the original T, T is transformed with T_s as the subject. For example, suppose that we are given the following T and H:

- T: 自激漏（じげきろう）は、1434年に中世李氏朝鮮の科学者、蒋英実が作った水時計である。[Cheugugi (Jigekiro) was a water gauge made by the scientist in the medieval Rissi Joseon, Jang Yeong-sil.]

- H: 蒋英実は中世李氏朝鮮の科学者である。[Jang Yeong-sil is a scientist in the medieval Rissi Joseon.]

In this case, the subject alignment connects '蒋英実'(Jang Yeong-sil) in H with '蒋英実'(Jang Yeong-sil) in T. Then, the transformation yields the graph containing several subtrees. If we extract such subtrees, this becomes the following four subtrees in T.

- T_1: <person> 蒋英実 (Jang Yeong-sil)</person> は、<job> 科学者 (scientist)</job> である。

[1] This irrelevant information is necessary in order to prune out from the cost calculation. This information should not aim at converting all the calculation between H and T into the irrelevant information, which is easily achievable.

- T_2: <person> 蒋英実 (Jang Yeong-sil)</person> は、<country> 中世李氏朝鮮 (the medieval Rissi Joseon)</country> の <job> 科学者 </job> である。

- T_3: <person> 蒋英実 (Jang Yeong-sil)</person> は、<time>1434 年 </time> に <object> 水時計 (water gauge)</object> を作った。

- T_4: coordination [<phrase> 自激漏 (Cheugugi)</phrase>, <phrase> じげきろう </phrase>]

- H: <person topic='Y'> 蒋英実 (Jang Yeong-sil)</person> は <country> 中世李氏朝鮮 (the medieval Rissi Joseon)</country> の <job> 科学者 (scientist)</job> である。

Hence, the graph matching algorithm eventually calculates the cost mostly between H and T_2. In this way, we modify the position of subject in a sentence according with H.

Coordination We obtained the relation of coordination from the dependency hypergraph.

Coreference Resolution / Identification of Non-anaphoric NPs In coreference resolution, the non-anaphoric definite NPs [10] are often given, but in our context they should be identified in its preparation. This should be also true for relative pronouns, reflexive pronouns, personal pronouns as well. Note that since there is no article in Japanese we have no distinction between whether definite NPs and nondefinite NPs. We identify the nonanaphoric NPs.

Coreference Resolution of Space / Time References We employ the space and time coreference resolution to identify the fluctuation of space and time expressions.

4.1.2 Unknown Words and Named Entities

Unknown Words (OOV Words) and Named Entities (Multi-Word Expressions) Unknown words or out-of-vocabulary words (OOV words) have negative overall effects in textual entailment task. We avoid this by searching them on the Internet resources. Similarly, the unknown named entities, such as person name, company names, and titles, may considerably decrease the overall performance. We use the Internet resources as well to find a possibly correct named entities (Multi-Word Expressions). Note that name can be written in various ways. For example, Leonald Da Vinci is equivalent with "Da Vinci", "Mr. Leonald Da Vince", "Leonald", and so forth.

4.1.3 Semantic Redundancies

Parenthesis and Quotation It is often that some phrases are emphasized or rephrased using parenthesis, quotation, and other symbols such as " 『』 ", " （） ", " 「」 ", """", and " " ". These expressions are considered semantically redundant, which is meaningful in understanding the text. For example, "The 8th 『This Mystery is amazing!』 prize" can be considered as one entity rather than only considering "This Mystery is amazing!".

Noisy Characters The text between parentheses and quotes may include noisy characters, which are somewhat superfluous in order to understanding the text. For example, "The 8th 'This Mystery is amazing!' prize" includes "!" and """", which are considered to be superfluous.

4.1.4 Text Understanding

A graph-matching-based textual entailment [2; 3] has limitation in that they will not detect whether T requires to understand the content of H. Suppose that T='Bob bought a red car from Mary and Tom.' and H='Three persons are related to the conversation'. In T, there is no number appeared, but human beings can read this sentence and understand that there are three persons in T. Our system considered the text understanding in terms of time / location / calculation.

Hypernym and Antonym This example is an usual situation for many literature. In a word level, if there is some difference in terms of the level of abstraction in two words, i.e. 中高年 and 中高生, it is required to judge whether 中高年 is a hypernym of 中高生 or these two does not have such relationships. Such relationships in word can be judge using lexical resources such as (Japanese) WordNet [11].

Quantifier Detection In English sentence, a quantifier, such as 'all' and 'every', needs to be examined in order to grasp the correct meaning. These are detected by the predefined vocabularies.

Rhetoric Detection If the sentence includes rhetoric, such as metaphor, prosopopoeia, and the idiomatic expression such as the four-word Kanji (i.e. “温故知新” and “南船北馬”), this may prevent the similarity-based matching approach. It is often that the title becomes rhetoric, such as in the case of “『Hey Hey おおきに毎度あり』”, in the sense that even if the meaning in the title matches with the surrounding meaning, it does not mean that T entails H. The text within “『』” should be considered to be a different layer of meaning.

4.2 Determination Step

The determination step judges the similarity of the T_i and H_i by the SVM classification [12] where T_i and H_i are the possible correspondent fragments. As is mentioned above, the feature extraction for the SVM classification algorithm are applied for the selected H_i and T_j. We used L1-loss function with Radial Basis Function (RBF) kernel where C and γ were determined by cross-validation. Major features which we used in our system are described below.

Lexical Entailment / Hyponymy Relations / Antonymy Relations / Location Relations / Adjective Gradation Features These features are the same as [13] and [3]. Note that depending on the deep / shallow linguistic preprocessing, hyponymy relations and antonymy relations are exchanged.

Modality / Polarity / Factivity Features These features capture the contexts which reverse or block monotonicity [3] where these are often marked by the presence or absence of linguistic markers. Modality feature capture modal reasoning where possibility will not entail actuality. Factivity feature

Adjunct Feature This feature suggests the dropping or adding of syntactic adjuncts moving from T to H [3].

Quantifier Features These features captures entailment relations among sentences involving quantification [3].

Semantic Role Matching Feature This feature indicates whether the corresponding semantic role relations are equivalent or not. As with this feature, some pair of features are preprocessed to give true or false beforehand.

Parenthesis and Quotation Features These features indicate the presence or absence of possible equivalent expressions. This enables the similarity matching with the expression among parenthesis and quotation.

Noisy Character / Rhetoric Feature These features suggest to drop the corresponding fragments from the similarity matching.

Time / Date / Number Features The presence of these features can be preprocessed by coreference resolution of space / time references (or some localization software) which will detect different form of equivalent expressions. These features are often preprocessed beforehand whether they are true of false.

Text Understanding Features Classification can only capture the similar expressions between T_i and H_j. As is mentioned in Chapter 3, when H_j requests some capability of text understanding of T_i, this feature would suggest some basic inference results in the deep / shallow linguistic preprocessing. This enables a judge whether T_i can be entailed H_j. Note that the capability of these features are limited in time, location, and number and in very basic case.

Content Length Feature If H contains more information than T, this can be immediately decided that T does not entail to H.

Deep Learning LM / Genre ID Feature Context- dependent language model feature is derived by context-dependent recurrent neural network language model [14]. Genre ID feature is derived by Latent Dirichlet Allocation (LDA) [15].

5 Experimental Settings and Results

	RTE2			RTE		
	Yes	No	total	Yes	No	total
JA dev	240	371	611	250	250	500
JA test	256	354	610	250	250	500
LGM-Y	175	31	206	191	63	254
LGM-N	81	323	404	59	187	246

Table 1. RTE2 set (left) and RTE set (right).

In the experiments, we used various deep / shallow linguistic preprocessing tools as well as resources, which are shown below: Morphological analyzer: JUMAN [16], Dependency parser: KNP [17], Named-entity recognizer, NLTK [18], MALLET [15], Paraphrase generator [19]; bootstrap method ("X deploy Y"), Wordnet [11], Wiki, monolingual corpora, and parallel corpora, Internet search engine: Google; Yahoo, Deep learning component: context-dependent recurrent neural network language model [14], ngram-HMM language model [20].

The statistics of development and test set for textual entailment BC task is shown in Table 1. The result by our approach is shown in Table 2 for RTE2 set (NTCIR-10) and Table 3 for RTE set (NTCIR-9). For RTE2 set, the Macro F1 score was 80.49. The precision for yes entailment was high, while the recall for no entailment was high. As is indicated by the row of our submission, our textual entailment system gave output of 'yes' with much smaller number than the correct answer, while it gave output of 'no' in larger number. The improvement from graph matching (GM1) and local graph matching (LGM) was 25% for RTE2 set. This tendency is kept for RTE as well. The improvement from graph matching (GM1) and local graph matching (LGM) was 20% for RTE set.

	LGM Best	GM 1	GM 2
Accuracy	81.64	55.16	65.16
Y-F1	75.76	50.18	58.46
Y-Prec	84.95	46.36	57.58
Y-Rec	68.36	54.69	59.38
N-F1	85.22	59.24	70.00
N-Prec	79.95	55.49	69.23
N-Rec	91.24	63.52	70.79

Table 2. RTE2 Testset: Table compares our method (LGM) and the original graph matching method. GM 1 is an original, while GM 2 does unknown words / parenthesis / space-time resolution.

6 Conclusion

This paper presented a local graph matching-based system with active learning. Our result for RTE2 corpus was 80.49 for macro F1 score, 84.95 for precision for the positive entailment, and 79.95 for recall for negative entailment. The reason for high precision for the positive entailment may be due to the fact that we tried to determine the entailment only when the distance between T' and H becomes small depending on deep / shallow linguistic preprocessing and determination.

	LGM Best	GM 1	GM 2
Accuracy	75.60	52.00	62.80
Y-F1	75.20	50.41	61.41
Y-Prec	74.49	48.80	59.20
Y-Rec	76.69	52.13	63.79
N-F1	75.88	53.49	64.09
N-Prec	73.86	51.88	61.94
N-Rec	78.00	55.20	66.40

Table 3. The results for RTE1 testset. LGM shows our method. GM 1 is an original, while GM 2 does unknown words / parenthesis / space-time resolution.

There are several avenues for further works. First, we briefly mentioned that the preprocessing is equivalent to reduce the dimensionality of matrices of $\text{Cost}_N(M)$ and $\text{Cost}_R(M)$. The relation extraction of [21] is attractive in this direction but requires to expand considerably. Second, we would like to extend this framework for crosslingual textual entailment and SMT [22]. Third, since T and H are traditionally include a lot of unnecessary elements we call them noisy elements. However, in SMT, T and H will be close. More subtle investigation of noisy element would be necessary [23; 24; 25].

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of CNGL at Dublin City University.

References

[1] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. *In Proceedings of the First PASCAL Recognizing Textual Entailment Workshop*, 2006.

[2] Aria Haghighi, Andrew Ng, and Christopher D. Manning. Robust textual inference via graph matching. *EMNLP*, 2005.

[3] Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. Learning to recognize features of valid textual entailments. *Proceedings of HLT-NAACL*, 2008.

[4] Shachar Mirkin, Ido Dagan, and Eyal Shnarch. Evaluating the inferential utility of lexical-semantic resources. *ACL-IJCNLP SRW*, 2009.

[5] David A. Forsyth and Jean Ponce. Computer vision. *Pearson International*, 2003.

[6] Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. Considering discourse references in textual entailment annotation. *5th International Conference on Generative Approaches to the Lexicon*, 2009.

[7] Christiane Fellbaum. Wordnet and wordnets. *Encyclopedia of Language and Linguistics Second Edition. Elsevier*, pages 665–670, 2005.

[8] Roy Rada, Mili Hafedh, Ellen Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on System, Man, and Cybernetics*, 19(1):17–30, 1989.

[9] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *In Proceedings of IJCAI*, 1995.

[10] D. Bean and E. Riloff. Unsupervised learning of contextual role knowledge for coreference resolution. *HLT/NAACL*, 2004.

[11] Francis Bond, Hitoshi Isahara, Sanae Fujita, Kiyotaka Uchimoto, Takayuki Kuribayashi, and Kyoko Kanzaki. Enhancing the japanese wordnet. *In Proceedings of the 7th Workshop on Asian Language Resources*, 2009.

[12] Nello Cristianini and John Showe-Taylor. Introduction to support vector machines. *Cambridge University Press*, 2000.

[13] Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Cer, Daniel Ramage, Chloe Kiddon, and Christopher D. Manning. Aligning semantic graphs for textual inference and machine reading. *AAAI Spring Symposium at Stanford*, 2006.

[14] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. *MSR-TR-2012-92*, 2012.

[15] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. *http://mallet.cs.umass.edu*, 2002.

[16] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. Improvements of Japanese Morphological Analyzer JUMAN. *In Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 22–28, 1994.

[17] Sadao Kurohashi and Makoto Nagao. KN Parser : Japanese Dependency/Case Structure Analyzer. *In Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 48–55, 1994.

[18] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with python— analyzing text with the natural language toolkit. *O'Reilly Media*, 2009.

[19] Chris Callison-Burch, David Talbot, and Miles Osborne. Statistical machine translation with word- and sentence-aligned parallel corpora. *ACL*, pages 175–182, 2004.

[20] Phil Blunsom and Trevor Cohn. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. *ACL*, 2011.

[21] Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. Relation extraction with matrix factorization and universal schemas. *In Proceedings of HLT-NAACL 13*, 2013.

[22] Tsuyoshi Okita, Qun Liu, and Josef van Genabith. Shallow semantically-informed pbsmt and hpbsmt. *In Proceedings of the Workshop on Statistical Machine Translation*, 2013.

[23] Tsuyoshi Okita. Data cleaning for word alignment. *In Proceedings of ACL-IJCNLP SRW*, pages 72–80, 2009.

[24] Tsuyoshi Okita, Alfredo Maldonado Guerra, Yvette Graham, and Andy Way. Multi-Word Expression sensitive word alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access*, pages 1–8, 2010.

[25] Tsuyoshi Okita. Word alignment and smoothing method in statistical machine translation: Noise, prior knowledge and overfitting. *PhD thesis at Dublin City University*, 2011.