含意関係パターンペアの大規模獲得
Large-scale acquisition of entailment pattern pairs

クロエツェー ジュリアン [†] Julien Kloetzer
鳥澤 健太郎 [†] Kentaro Torisawa
デ・サーガ ステイン [†] Stijn De Saeger
佐野 大樹 [†] Motoki Sano
橋本 力 [†] Chikara Hashimoto
後藤 淳 [†] Jun Gotoh

Abstract

This paper presents a supervised method for recognizing entailment between patterns such as "$Y_{\text{病気}}$ に効く $X_{\text{薬}}$" ("X_{drug} *efficient against* Y_{disease}"). Using distributional similarity features as well as surface features and lexical resources, we were able to acquire 185 million such pairs in Japanese with a precision of 80% from our 600 million pages web archive.

1 Introduction

Recognizing textual entailment has become an important topic in Natural Language Processing (NLP) since its introduction by Dagan et al. (2006). We say a text L entails a second text R if a human who reads L would be able to infer that R is true. For example, "珊瑚にダメージを与えているのだ" ("*causes damage to coral*") entails "珊瑚がダメージを受けるのだ" ("*Coral suffers from damage*"), but it does not entail "珊瑚をダメージから守る" ("*protect coral from damage*"). Recognizing textual entailment has a variety of applications in NLP, such as Question-Answering, summarization, information extraction and evaluation of machine translation.

Techniques for recognizing textual entailment between sentences vary from using logical representations of text [10, 17] to semantic analysis [3] or syntactic parsing [1, 18]. While all of these works deal with the recognition of entailment between two sentences or two text fragments, other works focus on the acquisition of entailment pairs, usually unary or binary patterns [2, 6, 9, 13]. Such pattern pairs can then be used later in a full-fledged system to recognize entailment between sentences [14] or to answer factoid questions [16].

Our work focuses on the acquisition of entailment pairs between typed lexico-syntactic binary patterns such as "$X_{\text{生物}}$ に $Y_{\text{損害}}$ を与えているのだ" ("*Causes* Y_{harm} *to* $X_{\text{life form}}$") or "$Y_{\text{病気}}$ に効く $X_{\text{薬}}$" ("X_{drug} *efficient against* Y_{disease}"). The subscripts "生物" ("*life form*") and "薬" ("*drug*") are examples of *types* that restrict the set of nouns which can fill the variables slots, and we say a pattern p entails a pattern q if for any noun-pairs that can fill the variables slots of p with the correct types one can deduce the information given by q with the same noun-pair. Our target is a set of 11 billion such typed pattern pairs in Japanese extracted from our 600 million web pages corpus. We wish to extract from this set highly precise entailment pairs to use into the factoid QA module of our large-scale Web information analysis system, WISDOM 2013 [16].

Acquisition of entailment pattern pairs usually exploits distributional similarity based scores to classify pattern pairs into entailment or non-entailment [6, 9, 19]. However, such scores did not give a strong performance on our target set. The main reason was that most patterns have low frequency because of the extensive coverage of this data as well as the restriction on typed patterns and, as such, distributional similarity based scores are less reliable.

We show in this paper that we can build a strong classifier for entailment between patterns by combining these distributional similarity scores with surface features and lexical resources, inspired by systems built to recognize entailment between sentences [11]. We built training data for this classifier by automatically choosing among various sets of hand-labeled pattern pairs an optimal combination of sets, and we trained a classifier which extracted from our candidate pattern pairs set around 160 million pattern pairs with a precision for entailment of above 80%. We also propose a classification of pattern pairs which we exploited to train more specific classifiers and extended our result to 185 million pattern pairs with about 80% precision. To the best of our knowledge, this is the biggest entailment data built for Japanese to this day.

The remaining of this paper is organized as follows: Section 2 details how we extracted the pattern pairs set we used as classification target. Section 3 explains how we built the training data for our classifier. In Section 4, we detail the feature set used by our classifier. Section 5 presents experimental results as well as our proposed pattern pairs classification. A conclusion follows in Section 6.

2 Target data

In this section, we explain how we extracted from our corpus of 600 million Japanese web pages a set of 11 billion candidate typed binary pattern pairs to be classified as entailment and non-entailment pairs.

[†] 情報通信研究機構 ユニバーサルコミュニケーション研究所 情報分析研究室
National Institute of Information and Communications Technology, Universal Communication Research Institute, Information Analysis Laboratory

2.1 Patterns

We extracted lexico-syntactic patterns from a corpus of 600 million web pages parsed with KNP[1]. In this work, patterns consist of words on the path of dependency relations connecting two nouns in a single sentence. We also restricted our patterns to those that co-occur with at least 10 different noun-pairs in our corpus and are at most 10 *bunsetsu*[2] long. We obtained this way 63 million unique patterns and their co-occurring noun pairs.

2.2 Typed patterns

As mentioned in the introduction, we focus on typed patterns, which are lexico-syntactic patterns that place semantic class restrictions on the noun pairs they co-occur with, such as "$X_{\text{生物}}$ に $Y_{\text{損害}}$ を与えているのだ" ("*Causes* Y_{harm} *to* $X_{\text{life form}}$"). The subscripts "損害" ("*harm*") and "生物" ("*life form*") specify the permitted semantic classes of the X and Y slot fillers. As shown in past works [2, 4], typed patterns make it possible to distinguish between multiple senses of ambiguous patterns, thus greatly reducing errors due to pattern ambiguity. For instance, the typed pattern "$X_{\text{損害}}$ に $Y_{\text{生物}}$ を与えているのだ" ("*Causes* X_{harm} *to* $Y_{\text{life form}}$") entails that "$X_{\text{生物}}$ が $Y_{\text{損害}}$ を受けるのだ" ("$X_{\text{life form}}$ *suffers from* Y_{harm}"), which may not hold for a differently typed version of the same pattern such as "$X_{\text{人}}$ に $Y_{\text{感じ}}$ を与えているのだ" ("*Causes* Y_{emotion} *to* X_{person}").

As a source of semantic classes for pattern typing, previous works have mainly used named entity recognizers or lexical resources such as WordNet [12]. We feel this limits the coverage of typed patterns, so in this work we follow De Saeger et al. [4] in inducing semantic classes automatically from our corpus. For this we use the EM based noun clustering algorithm presented by Kazama et al. [7], which computes the probability that a word w belongs to a hidden class c, i.e., $P(c|w)$. We clustered 1 million nouns into 500 semantic classes, which gives 250,000 possible semantic class combinations for typing patterns. Using these class pairs, we obtained a total of 2.8 billion typed patterns cooccuring with at least one noun-pair in their given semantic class pair. From here on, and unless specified otherwise, we will always consider patterns or relations between patterns in the context of some semantic class pair.

2.3 Entailment candidate pattern pairs for classification

Since two patterns taken randomly have a very low probability of having an entailment relation, even when limited to patterns in the same semantic class pair, we decided to restrict our candidate pattern pairs for classification to the set *TARGET* of pairs of patterns which co-occur with at least three common distinct noun pairs. While this restriction may seem strict, there are still more than 11 billion pattern pairs to classify in set *TARGET*. Set *TARGET* contains approximately 7% of pairs with an entailment relation (about 770 million entailment pairs).

3 Training data

In this section, we explain how we built sets of hand-labeled data for entailment, and then how we selected the most appropriate sets of labeled data to use as training data for our classifier.

3.1 Annotated data sets

In the course of this research and of other works related to pattern entailment, we have built multiple hand-labeled entailment pair sets: we now have 11 sets of pattern pairs whose size range from 500 pattern pairs to 22,500 pattern pairs, for a total of 74,000 pattern pairs. 10 out of these sets consist of pattern pairs randomly selected from *TARGET* but with different restrictions each, like for example having high distributional similarity scores or having some specific kind of verb pairs. The last set is not restricted to *TARGET*, but all of its (single) patterns appear in our corpus. All of these pattern pairs were annotated by three human annotators as entailment or non-entailment. We considered a pattern pair as a true entailment relation if at least two out of the three annotators marked it as positive, and as non-entailment otherwise. The inter-rater agreement score (Fleiss Kappa) for the whole of the pattern pairs was 0.67, indicating substantial agreement [8].

We say binary patterns $\langle p, q \rangle$ such as ⟨"X に Y を与えているのだ", "X が Y を受けるのだ"⟩ (⟨"*causes Y to X*", "*X suffers from Y*"⟩) have an entailment relation if a human reader reading p is able to infer the information given by q for any noun pair that can instantiate the patterns' variables in the provided semantic class pair. Because our semantic classes are obtained by automatic clustering and have no meaningful labels, we followed Szpektor et al. [15] and provided the annotators with three random noun pairs that co-occur with the first pattern of a pair as a proxy for the class pair. The annotators marked a given pattern pair as positive if the entailment relation between the patterns held for all three noun pairs presented.

3.2 Training data selection

While most of our hand-labeled data sets are extracted from *TARGET*, they usually make bad representatives of *TARGET* because their pattern pairs were not sampled randomly. For this reason, we believe that using all of them as training data may overall hurt our classifier and hence devised the following process to select an optimal set of pattern pairs to use a training data:

1. Select one set to be used as *TEST* data.

2. Generate every possible combination of the remaining pattern pairs sets: these combination of sets are used as candidate training data sets (for examples, for three sets A, B and C, the candidate training data sets would be the seven sets A, B, C, $A \cup B$, $B \cup C$, $A \cup C$ and $A \cup B \cup C$).

3. Train a classifier for each candidate training data set.

[1] Kurohashi-Nagao Parser, `http://nlp.ist.i.kyotou.ac.jp/EN/index.php?KNP`

[2] bunsetsu: smallest unit of words that sounds natural in a sentence

4. Evaluate each classifier by the average precision it obtains when classifying *TEST*, and choose the classifier that obtains the best average precision.

Plotting precision $P(r)$ as a function of recall r obtained when ranking set *TEST*, the average precision over set *TEST* is defined as follows:

$$AveP = \sum_{k=1}^{n} P(k)\delta r(k) \quad (1)$$

Here, k is the rank in the sequence of samples of *TEST*, n is the number of elements in *TEST*, $P(k)$ is the precision at cut-off k in the list, and $\delta r(k)$ is the change in recall from items $k-1$ to k.

In extreme cases, average precision may rank low a classifier with high precision for low recall and low precision otherwise, even though such a classifier's top ranked pairs may be of very high quality. We realize that this ranking of classifiers using average precision may not be the best way to select a classifier, especially since this formula does not take at all into account local maxima, but such extreme cases are rare and we believe that this ranking is fair overall.

Since we wanted to optimize our classifier for classifying *TARGET*, we used as set *TEST* in our experiments a set of 5000 hand labeled pairs randomly sampled from *TARGET*. We then tried the 1023 possible combinations of the 10 remaining sets using the above procedure and, for the combination with the highest average precision, we also tried adding set *TEST* as training data by making a 10-fold cross validation experiment (10-fold cross validation being 10 times more time expensive, we did not try it for every possible combination of training data). The best classifier we obtained has training data consisting of the combination of 8 data sets (including *TEST*), for a total of 67,000 training samples. We call this training data set *TRAIN*.

4 Features set

In this section, we present the three types of features used for our classifier. The first type is surface features, which consider n-grams of characters or morphemes extracted from the patterns themselves and measure their similarity on the surface level. The second type is lexical resources related features, which signal the presence of some words or pairs of words in the pattern pair in pre-computed word or word pair databases such as pairs of synonyms or antonyms. Finally, the third type is distributional similarity based features, which measure the similarity between patterns in term of the context they appear in, that is the set of noun pairs they co-occur with. All of these features are summarized in Table 1.

4.1 Surface features

We followed the work of Malakasiotis et. al [11] to design this feature set. They propose, to recognize entailment between sentences, to combine different similarity measures computed using bag-of-words representations of each sentence. The underlying idea is that while a single similarity measure can surely not solve the whole problem, combining them in a classifier should provide enough evidence to judge for an entailment relation.

To compute these features for a given pattern pair $\langle p, q\rangle$, we first compute for each pattern the following bag-of-words: (1) the sets of 1-, 2- and 3-grams of the pattern's characters, (2) the sets of 1-, 2- and 3-grams of the pattern's morphemes, (3) stem versions of (2), (4) the n-grams sets of POS tags of (2), and (5) the sets of verbs, adjectives and nouns of the pattern (separately) as well as their stems, and (6) the sets of sub-trees extracted from each pattern's dependency tree.

Then, for each pair of bag-of-words $\langle P, Q\rangle$ extracted from p and q, we compute the following measures and scores: (a) the cardinal of each set (two values), (b) the cardinal of the sets intersection, (c) the ratio of elements in each set also in the other (two values), (d) the pair's Dice Coefficient, and (e) the pair's Jaccard score as well as a variant of this score, discounted Jaccard score.

Representing the bag-of-words as multi-sets (i.e., frequency vectors), we also compute the following distances between P and Q: (f) the cosine distance, (g) the Manhattan distance, (h) the Euclidian distance, and (i) a frequency based variant of the Jaccard score.

Finally, considering the bag-of-words as ordered sets (in the order in which the words or n-grams come in the patterns), we compute the following two distances: (j) the Levenshtein distance, and (k) the Jaro distance.

We also include the following as features: binary features signalling the presence of each of the patterns 1-grams and 2-grams and each content-word pair extracted from the pattern pair (content words are nouns, verbs and adjectives), as well as the patterns length ratios.

4.2 Lexical resources features

Surface features can detect simple syntactical variations between patterns but lack higher level knowledge such as synonymy or allography. To solve this problem, we integrate a number of word pair databases in our classifier's feature set.

We first extract from a pattern pair $\langle p, q\rangle$ the set of word pairs $\langle wp, wq\rangle$ such that wp is in p and wq is in q. Then, for each word-pair database, we set a feature to 1 if any word pair $\langle wp, wq\rangle$ extracted from $\langle p, q\rangle$ is in this database. We also do the same for the stems of the words, and for the pairs $\langle wp, wq\rangle$ (some databases are directional, so we want to detect relations both ways).

We use the following word pair databases: (a) 4 databases of entailment verb pairs , (b) 4 databases of non-entailment verb pairs, (c) 2 databases of allographic words and (d) databases of synonyms, (e) antonyms and (f) part-of word pairs, all of (a) to (f) available at the ALAGIN forum[3] (reference code A-2, A-7 and A-9), (g) a database of allographic words and (h) a database of antonyms, both extracted from the morphological analyzer JUMAN's dictionary[4], (i) 16 databases of contradictory word pairs and template pairs from a precedent research [5], and finally (j) the transitive closure up to 7 steps obtained when combining (a), (c), (d) and (g).

[3] http://www.alagin.jp/

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

Table 1: Features summary, computed over a pair of patterns $\langle p, q \rangle$

surface	**Similarity measures:** common elements ratios, Dice coefficient, Jaccard and discounted Jaccard scores, Cosine, Euclidian, Manhattan, Levenshtein and Jaro distances; *computed over:* the patterns' 1-, 2- and 3-grams sets of: characters, morphemes, their stems & POS; content words and stems
	binary feature for each of the patterns' subtrees, 1- and 2-grams ; patterns' lengths and length ratios
Lexic.	Entries in databases of verb entailments and non-entailments, synonyms, antonyms, allographs ; *checked over:* pairs of content words, pairs of content word stems, same for the reverse pattern pair $\langle q, p \rangle$
dis. sim.	**Distributional similarity measures:** Common elements ratios, Jaccard and discounted Jaccard scores, sets and sets intersection cardinality, DIRT [9], Weeds [19] and Hashimoto [6] scores; *computed over*: patterns' co-occurring noun pairs, POS tags of those, nouns co-occurring in each variable slot and with each unary sub-patterns
	binary feature for each semantic class pair and individual classes, patterns frequency rank in the given class pair

Following the work on Excitation of [5], we also integrate (k) 5 databases of unary patterns hand-labeled as excitatory, inhibitory, neutral, ungrammatical or undecided. Since those databases consists of single unary patterns (not pairs like the previous ones), we have for each database a feature for each pattern noting the presence of one of the database's elements in the pattern.

4.3 Distributional similarity features

Distributional similarity measures are based on the idea that phrases with similar meaning tend to appear into similar contexts. In the case of binary patterns, these measures are computed over the sets of noun-pairs co-occurring with each pattern, based on the intuition that patterns that co-occur with similar noun-pairs should have similar meanings. While these measures were at first proposed to detect paraphrases [9], that is bi-directional entailment, some recently proposed measures are directional and specifically used to detect entailment [6, 19].

To compute these features for a given pattern $\langle p, q \rangle$, we first compute for each pattern the following sets: (a) the set of noun-pairs that co-occur with the pattern in our corpus, (b) the two sets of nouns that co-occur with the pattern in each variable slot, (c) the sets of POS-tags of each set of (b), and (d) the two sets of nouns that co-occur with each of the pattern's unary sub-patterns in our corpus in the case where each of the binary patterns p and q can be decomposed into two unary sub-patterns, one attached to each variable.

Then, for each of these sets pairs $\langle P, Q \rangle$, we compute the following scores: (a) the cardinal of each set (two values), (b) the cardinal of the sets intersection, (c) the ratio of elements in each set also in the other (two values), (d) the pair's Dice Coefficient, and (e) the pair's Jaccard score as well as a variant of this score, discounted Jaccard score.

Also, representing the sets as multi-sets (i.e., frequency vectors), we also compute the following measures between P and Q: (f) DIRT score [9], (g) Weeds' precision [19], (h) Hashimoto score [6] (we also compute (g) and (h) for the pair $\langle Q, P \rangle$, since those scores are directional), and (i) a frequency based variant of the Jaccard score.

Finally, although these are not technically distributional similarity measures, we included the following as related features: (α) the semantic classes in which the pattern pair is considered, and (β) the rank of each pattern in terms of number of co-occurring noun-pairs (lower-ranked patterns have more generic meaning).

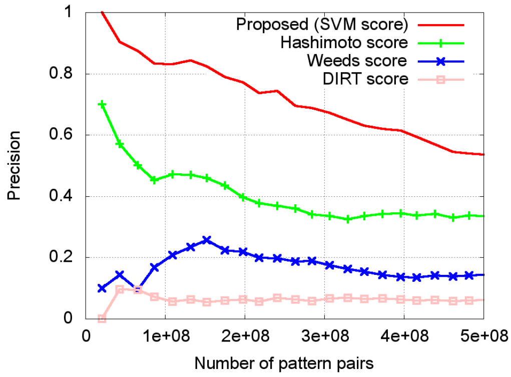

Figure 1: Precision curves for our method and three baselines (top 500 million pairs)

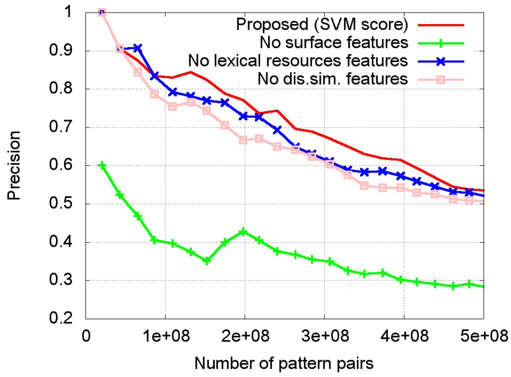

Figure 2: Ablation test (top 500 million pairs)

5 Experimental results

In this section, we present experimental results for our classifier as well as a categorization of the pattern pairs that we used to further enhance the results of our method.

5.1 Basic Classification results

We trained a classifier using the training data set *TRAIN* obtained as described in Section 3.2 and the software TinySVM[5] with a polynomial kernel of degree 2 (all experiments were done in this setting, which we decided from earlier experiments).

Using this classifier, we classified the 5000 pairs of set *TEST* and ranked them by the score given by the classifier. Assuming that these pairs are distributed uniformly over

[5] `http://chasen.org/~taku/software/TinySVM/`

Table 2: Examples of acquired pattern pairs (with an example of noun-pair)

Entailing pattern	Entailed pattern	SVM score	Category
X(画像) に Y(字) を載せる Put Y(characters) on the X(video)	X(画像) に Y(字) を添える Attach Y(characters) to the Y(video)	0.737	Other

X(カラオケ) で Y(練習) を始めます
Start Y(training) at the X(karaoke)
	Y(練習) の X(カラオケ)
	Y(training)'s X(karaoke)
		0.494	No content

X(部屋) で Y(エアコン) をつけている the Y(air-conditioner) is set in the X(room)	X(部屋) で Y(エアコン) つけてる set the Y(air conditioner) in the X(room)	0.346	Dimilar

X(画像) が貼られた Y(ブログ)
X(videos) attached to the Y(blog)
	X(画像) を展示する Y(ブログ)
	the Y(blog) displays X(videos)
		0.153	DB covered

X(珊瑚) に Y(ダメージ) を与えているのだ causes Y(damage) to X(coral)	X(珊瑚) が Y(ダメージ) を受けるのだ X(coral) suffers from Y(damage)	0.151	DB covered
Y(Ｓｉｔｅ) を始めた X(理由) The X(reason) I started the Y(site)	Y(Ｓｉｔｅ) をしている X(理由) The X(reason) I am doing the Y(site)	0.045	Other

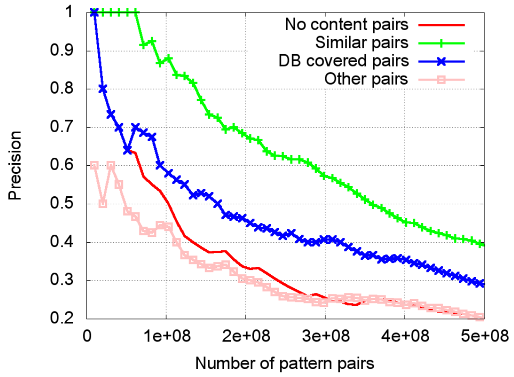

Figure 3: Precision curves for each pair category (top 500 million pairs)

the ranking of *TARGET*, we drew in Figure 1 the precision curve obtained by the classifier. For comparison purposes, we also show the precision curves obtained when ranking *TEST* pairs by three unsupervised baselines: Hashimoto score [6], Weeds precision [19] and DIRT score [9]. We also drew in Figure 2 the precision curves obtained by our classifier when using all the features, and when removing each feature type (surface, lexical resources, distributional similarity) in turn, for an ablation test.

According to these results, the top 160 million pattern pairs ranked by our classifier have a precision of about 80%. Table 2 presents examples of pairs acquired by our classifier. Also, as shown by the performance obtained by the distributional similarity based baselines, these cannot compete with a supervised classifier in such a setting, although the performance of the Hashimoto score is strong for an unsupervised baseline. Finally, the ablation test shows that all types of features are necessary to obtain a strong classifier, although the surface features are the most prevalent of all.

5.2 Pattern pair categorization and enhanced classification results

By analyzing the output of our classifier, we found it is possible to classify pattern pairs into 4 categories according to the patterns' content words (here, a content word is a noun, a verb or an adjective):

- "*No content*" pattern pairs, where at least one of the patterns has no content word, like for example "*A* の *B*" ("*A's B*").
- "*Similar*" pattern pairs, where both patterns have at least one content word in common.
- "*Database covered*" pattern pairs, where one of the pattern pair's content word pair is found into one of the word pair databases we used in for our feature set.
- "*Other*" pattern pairs, which do not fit into any of the previous three categories.

Using the same procedure described in Section 3.2, only this time restricting test set *TEST* to pattern pairs of each of the 4 categories in turn, we obtained 4 classifiers which performance (in terms of average precision) is optimal when classifying pairs of set *TEST* for each of the 4 categories. Because they are optimal, these 4 classifiers perform in their own categories at least better than the single classifier we presented in the previous section. Hence, we were able to improve the overall output of our method by classifying each pattern pair of *TARGET* using the best classifier for the pair's category.

We drew in Figure 3 the precision curves obtained for the four classifiers optimized for each of the four categories. Clearly as expected, "*Similar*" pattern pairs are the easiest to classify and those for which our method performs best. Also, while "*Database covered*" pairs have strong evidence showing their entailment/non-entailment in terms of a pair of content words, the performance of our best classifier for these pairs is disappointing. Finally, "*Other*" pairs are, as expected, the most challenging. By combining the output of these four classifiers whith a precision of 80%, we obtained in total 185 million pairs with 80% precision for entailment. This data is the final output of our work.

6 Conclusion

In this paper, we present the supervised method we used to classify 11 billion pattern pairs into entailment and non-entailment pairs. By combining distributional similarity measures with surface features and lexical resources and automatically combining hand-labeled data sets to generate our classifier's training data, we were able to obtain a set of 185 million pattern pairs with precision of about 80%. We plan to release this data through the ALAGIN forum.

References

[1] R. Bar Haim, J. Berant, I. Dagan, I. Greental, S. Mirkin, E. Shnarch, and I. Szpektor. Efficient semantic deduction and approximate matching over compact parse forests. In *Proceedings of Text Analysis Conference*, 2009.

[2] J. Berant, I. Dagan, and J. Goldberger. Global learning of typed entailment rules. In *Proceedings of ACL 2011*, pages 610–619, 2011.

[3] Aljoscha Burchardt, Nils Reiter, Stefan Thater, Anette Frank, and Dept Computational Linguistics. A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[4] S. De Saeger, K. Torisawa, J. Kazama, K. Kuroda, and M. Murata. Large scale relation acquisition using class dependent patterns. In *Proceedings of ICDM 2009*, page 764—769, 2009.

[5] C. Hashimoto, K. Torisawa, S. De Saeger, J.-H. Oh, and J. Kazama. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In *Proceedings of EMNLP 2012*, 2012.

[6] C. Hashimoto, K. Torisawa, K. Kuroda, S. De Saeger, M. Murata, and J. Kazama. Large-scale verb entailment acquisition from the web. In *Proceedings of EMNLP 2009*, volume 3, page 1172—1181, 2009.

[7] J. Kazama and K. Torisawa. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. *Proceedings of ACL 2008*, page 407—415, 2008.

[8] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, page 159—174, 1977.

[9] D. Lin and P. Pantel. Dirt - discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, 2001.

[10] Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, page 193—200, 2007.

[11] P. Malakasiotis and I. Androutsopoulos. Learning textual entailment using SVMs and string similarity measures. In *Proceedings of the ACL- PASCAL Workshop on Textual Entailment and Paraphrasing*, page 42—47, 2007.

[12] S. Schoenmackers, O. Etzioni, D. S Weld, and J. Davis. Learning first-order horn clauses from web text. In *Proceedings of EMNLP 2010*, page 1088—1098, 2010.

[13] Stefan Schoenmackers, Oren Etzioni, and Daniel S Weld. Scaling textual inference to the web. In *Proc. of the EMNLP2008*, page 79—88, 2008.

[14] Eyal Shnarch, Jacob Goldberger, and Ido Dagan. A probabilistic modeling framework for lexical entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, page 558—563, Portland, Oregon, 2011. Association for Computational Linguistics. ACM ID: 2002846.

[15] I. Szpektor, E. Shnarch, and I. Dagan. Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL 2007*, volume 45, page 456—463, 2007.

[16] Masahiro Tanaka, Stijn De Saeger, Kiyonori Ohtake, Chikara Hashimoto, Makoto Hijiya, Hideaki Fujii, and Kentaro Torisawa. Wisdom2013: A large-scale web information analysis system. In *Proceedings of the IJCNLP 2013*, page to appear, 2013.

[17] M. Tatu and D. Moldovan. COGEX at RTE3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, page 22—27, 2007.

[18] Rui Wang, Yi Zhang, and Guenter Neumann. A joint syntactic-semantic representation for recognizing textual relatedness. *Proc. of TAC*, 2009.

[19] J. Weeds and D. Weir. A general framework for distributional similarity. In *Proceedings of EMNLP 2003*, page 81—88. Association for Computational Linguistics, 2003.