Clustering Clauses for High-Level Relation Detection: An Information-theoretic Approach

Samuel Brody
School of Informatics
University of Edinburgh
s.brody@sms.ed.ac.uk

Abstract

Recently, there has been a rise of interest in unsupervised detection of high-level semantic relations involving complex units, such as phrases and whole sentences. Typically such approaches are faced with two main obstacles: data sparseness and correctly generalizing from the examples. In this work, we describe the *Clustered Clause* representation, which utilizes information-based clustering and inter-sentence dependencies to create a simplified and generalized representation of the grammatical clause. We implement an algorithm which uses this representation to detect a predefined set of high-level relations, and demonstrate our model's effectiveness in overcoming both the problems mentioned.

1 Introduction

The semantic relationship between words, and the extraction of meaning from syntactic data has been one of the main points of research in the field of computational linguistics (see Section 5 and references therein). Until recently, the focus has remained largely either at the single word or sentence level (for instance: dependency extraction, word-to-word semantic similarity from syntax, etc.) or on relations between units at a very high context level such as the entire paragraph or document (e.g. categorizing documents by topic).

Recently there have been several attempts to define frameworks for detecting and studying interactions at an intermediate context level, and involving whole clauses or sentences. Dagan et al. (2005) have emphasized the importance of detecting textual-entailment/implication between two sentences, and its place as a key component in many real-world applications, such as Information Retrieval and Question Answering.

When designing such a framework, one is faced with several obstacles. As we approach higher levels of complexity, the problem of defining the basic units we study (e.g. words, sentences etc.) and the increasing amount of interactions make the task very difficult. In addition, the data sparseness problem becomes more acute as the data units become more complex and have an increasing number of possible values, despite the fact that many of these values have similar or identical meaning.

In this paper we demonstrate an approach to solving the complexity and data sparseness problems in the task of detecting relations between sentences or clauses. We present the *Clustered Clause* structure, which utilizes information-based clustering and dependencies within the sentence to create a simplified and generalized representation of the grammatical clause and is designed to overcome both these problems.

The clustering method we employ is an integral part of the model. We evaluate our clusters against semantic similarity measures defined on the human-annotated WordNet structure (Fellbaum, 1998). The results of these comparisons show that our cluster members are very similar semantically. We also define a high-level relation detection task involving relations between clauses, evaluate our results, and demonstrate

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 448–455,
Prague, Czech Republic, June 2007.

the effectiveness of using our model in this task.

This work extends selected parts of Brody (2005), where further details can be found.

2 Model Construction

When designing our framework, we must address the complexity and sparseness problems encountered when dealing with whole sentences. Our solution to these issues combines two elements. First, to reduce complexity, we simplify a grammatical clause to its primary components - the subject, verb and object. Secondly, to provide a generalization framework which will enable us to overcome data-sparseness, we cluster each part of the clause using data from within the clause itself. By combining the simplified clause structure and the clustering we produce our *Clustered Clause* model - a triplet of clusters representing a generalized clause.

The Simplified Clause: In order to extract clauses from the text, we use Lin's parser MINIPAR (Lin, 1994). The output of the parser is a dependency tree of each sentence, also containing lemmatized versions of the component words. We extract the verb, subject and object of every clause (including subordinate clauses), and use this triplet of values, the simplified clause, in place of the original complete clause. As seen in Figure 1, these components make up the top (root) triangle of the clause parse tree. We also use the lemmatized form of the words provided by the parser, to further reduce complexity.

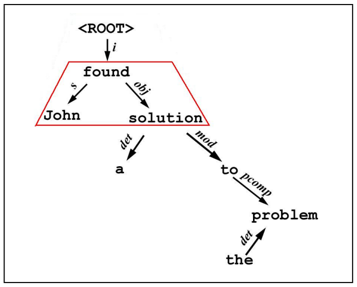

Figure 1: The parse tree for the sentence "John found a solution to the problem". The subject-verb-object triplet is marked with a border.

Clustering Clause Components: For our model, we cluster the data to provide both generalization, by using a cluster to represent a more generalized concept shared by its component words, and a form of dimensionality reduction, by using fewer units (clusters) to represent a much larger amount of words.

We chose to use the Sequential Information Bottleneck algorithm (Slonim et al., 2002) for our clustering tasks. The information Bottleneck principle views the clustering task as an optimization problem, where the clustering algorithm attempts to group together values of one variable while retaining as much information as possible regarding the values of another (target) variable. There is a trade-off between the compactness of the clustering and the amount of retained information. This algorithm (and others based on the IB principle) is especially suited for use with graphical models or dependency structures, since the distance measure it employs in the clustering is defined solely by the dependency relation between two variables, and therefore required no external parameters. The values of one variable are clustered using their co-occurrence distribution with regard to the values of the second (target) variable in the dependency relation. As an example, consider the following subject-verb co-occurrence matrix:

S \ V	**tell**	**scratch**	**drink**
dog	0	4	5
John	4	0	9
cat	0	6	3
man	6	1	2

The value in cell (i, j) indicates the number of times the noun i occurred as the subject of the verb j. Calculating the Mutual Information between the subjects variable (S) and verbs variable (V) in this table, we get $MI(S, V) = 0.52$ bits. Suppose we wish to cluster the subject nouns into two clusters while preserving the highest Mutual Information with regard to the verbs. The following co-occurrence matrix is the optimal clustering, and retains a M.I. value of 0.4 bits (77% of original):

Clustered S \ V	**tell**	**scratch**	**drink**
{dog,cat}	0	10	8
{John,man}	10	1	11

Notice that although the values in the *drink* column are higher than in others, and we may be tempted to cluster together *dog* and *John* based on this column, the informativeness of this verb is smaller - if we know the verb is *tell* we can be sure the noun is not *dog* or *cat*, whereas if we know it is *drink*, we can only say it is slightly more probable that the noun is *John* or *dog*.

Our dependency structure consists of three variables: subject, verb, and object, and we take advantage of the subject-verb and verb-object dependencies in our clustering. The clustering was performed on each variable separately, in a two phase procedure (see Figure 2). In the first stage, we clustered the subject variable into 200 clusters[1], using the subject-verb dependency (i.e. the verb variable was the target). The same was done with the object variable, using the verb-object dependency. In the second phase, we wish to cluster the verb values with regard to both the subject and object variables. We could not use all pairs of subjects and objects values as the target variable in this task, since too many such combinations exist. Instead, we used a variable composed of all the pairs of subject and object *clusters* as the target for the verb clustering. In this fashion we produced 100 verb clusters.

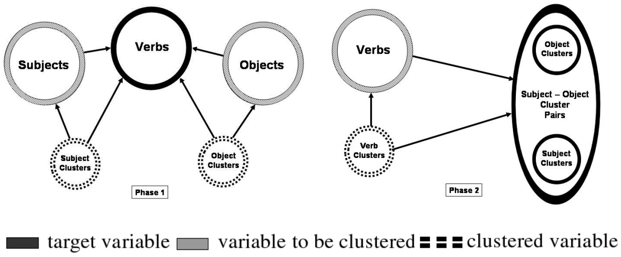

Figure 2: The two clustering phases. Arrows represent dependencies between the variables which are used in the clustering.

Combining the Model Elements: Having obtained our three clustered variables, our original simplified clause triplet can now be used to produce the *Clustered Clause* model. This model represents a clause in the data by a triplet of cluster indexes, one cluster index for each clustered variable. In order to map a clause in the text to its corresponding clustered clause, it is first parsed and lemmatized to obtain the subject, verb and object values, as described above, and then assigned to the clustered clause in which the subject cluster index is that of the cluster containing the subject word of the clause, and the same for the verb and object words. For example, the sentence “The terrorist threw the grenade” would be converted to the triplet (terrorist, throw, grenade) and assigned to the clustered clause composed of the three clusters to which these words belong. Other triplets assigned to this clustered clause might include (fundamentalist, throw, bomb) or (militant, toss, explosive). Applying this procedure to the entire text corpus results in a distillation of the text into a series of clustered clauses containing the essential information about the actions described in the text.

Technical Specifications: For this work we chose to use the entire Reuters Corpus (English, release 2000), containing 800,000 news articles collected uniformly from 20/8/1996 to 19/8/1997. Before clustering, several preprocessing steps were taken. We had a very large amount of word values for each of the Subject (85,563), Verb (4,593) and Object (74,842) grammatical categories. Many of the words were infrequent proper nouns or rare verbs and were of little interest in the pattern recognition task. We therefore removed the less frequent words - those appearing in their category less than one hundred times. We also cleaned our data by removing all words that were one letter in length, other than the word ‘I’. These were mostly initials in names of people or companies, which were uninformative without the surrounding context. This processing step brought us to the final count of 2,874,763 clause triplets (75.8% of the original number), containing 3,153 distinct subjects, 1,716 distinct verbs, and 3,312 distinct objects. These values were clustered as described above. The clusters were used to convert the simplified clauses into clustered clauses.

3 Evaluating Cluster Quality

Examples of some of the resulting clusters are provided in Table 1. When manually examin-

[1] The chosen numbers of clusters are such that each the resulting clustered variables preserved approximately half of the co-occurrence mutual information that existed between the original (unclustered) variable and its target.

“Technical Developements” (Subject Cluster 160): treatment, drug, method, tactic, version, technology, software, design, device, vaccine, ending, tool, mechanism, technique, instrument, therapy, concept, model
“Ideals/Virtues” (Object Cluster 14): sovereignty, dominance, logic, validity, legitimacy, freedom, discipline, viability, referendum, wisdom, innocence, credential, integrity, independence
“Emphasis Verbs” (Verb Cluster 92): imply, signify, highlight, mirror, exacerbate, mark, signal, underscore, compound, precipitate, mask, illustrate, herald, reinforce, suggest, underline, aggravate, reflect, demonstrate, spell, indicate, deepen
“Plans” (Object Cluster 33): journey, arrangement, trip, effort, attempt, revolution, pullout, handover, sweep, preparation, filing, start, play, repatriation, redeployment, landing, visit, push, transition, process

Table 1: Example clusters (labeled manually).

ing the clusters, we noticed the “fine-tuning” of some of the clusters. For instance, we had a cluster of countries involved in military conflicts, and another for other countries; a cluster for winning game scores, and another for ties; etc. The fact that the algorithm separated these clusters indicates that the distinction between them is important with regard to the interactions within the clause. For instance, in the first example, the context in which countries from the first cluster appear is very different from that involving countries in the second cluster.

The effect of the dependencies we use is also strongly felt. Many clusters can be described by labels such as “things that are thrown” (*rock, flower, bottle, grenade* and others), or “verbs describing attacks” (*spearhead, foil, intensify, mount, repulse* and others). While such criteria may not be the first choice of someone who is asked to cluster verbs or nouns, they represent unifying themes which are very appropriate to pattern detection tasks, in which we wish to detect connections between actions described in the clauses. For instance, we would like to detect the relation between throwing and military/police action (much of the throwing described in the news reports fits this relation). In order to do this, we must have clusters which unite the words relevant to those actions. Other criteria for clustering would most likely not be suitable, since they would probably not put *egg, bottle* and *rock* in the same category. In this respect, our clustering method provides a more effective modeling of the domain knowledge.

3.1 Evaluation via Semantic Resource

Since the success of our pattern detection task depends to a large extent on the quality of our clusters, we performed an experiment designed to evaluate semantic similarity between members of our clusters. For this purpose we made use of the WordNet Similarity package (Pedersen et al., 2004). This package contains many similarity measures, and we selected three of them (Resnik (1995), Leacock and Chodorow (1997), Hirst and St-Onge (1997)), which make use of different aspects of WordNet (hierarchy and graph structure). We measured the average pairwise similarity between any two words appearing in the same cluster. We then performed the same calculation on a random grouping of the words, and compared the two scores. The results (Fig. 3) show that our clustering, based on co-occurrence statistics and dependencies within the sentence, correlates with a purely semantic similarity as represented by the WordNet structure, and cannot be attributed to chance.

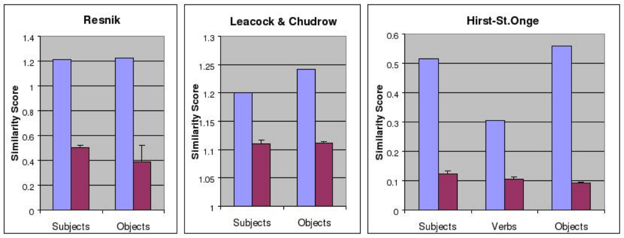

Figure 3: Inter-cluster similarity (average pairwise similarity between cluster members) in our clustering (light) and a random one (dark). Random clustering was performed 10 times. Average values are shown with error bars to indicate standard deviation. Only Hirst & St-Onge measure verb similarity.

4 Relation Detection Task

Motivation: In order to demonstrate the use of our model, we chose a relation detection task. The workshop on entailment mentioned in the introduction was mainly focused on detecting whether or not an entailment relation exists between two texts. In this work we present a complementary approach - a method designed to automatically detect relations between portions of text and generate a knowledge base of the detected relations in a generalized form. As stated by (Dagan et al., 2005), such relations are important for IR applications. In addition, the patterns we employ are likely to be useful in other linguistic tasks involving whole clauses, such as paraphrase acquisition.

Pattern Definition: For our relation detection task, we searched for instances of predefined patterns indicating a relation between two clustered clauses. We restricted the search to clause pairs which co-occur within a distance of ten clauses[2] from each other. In addition to the distance restriction, we required an *anchor*: a noun that appears in both clauses, to further strengthen the relation between them. Noun anchors establish the fact that the two component actions described by the pattern involve the same entities, implying a direct connection between them. The use of verb anchors was also tested, but found to be less helpful in detecting significant patterns, since in most cases it simply found verbs which tend to repeat themselves frequently in a context. The method we describe assumes that statistically significant co-occurrences indicate a relationship between the clauses, but does not attempt to determine the type of relation.

Significance Calculation: The patterns detected by the system were scored using the statistical *p-value* measure. This value represents the probability of detecting a certain number of occurrences of a given pattern in the data under the independence assumption, i.e. assuming there is no connection between the two halves of the pattern. If the system has detected k instances of a certain pattern, we calculate the probability of encountering this number of instances under the independence assumption. The smaller the probability, the higher the significance. We consider patterns with a chance probability lower than 5% to be significant.

We assume a Gaussian-like distribution of occurrence probability for each pattern[3]. In order to estimate the mean and standard deviation values, we created 100 simulated sequences of triplets (representing clustered clauses) which were independently distributed and varied only in their overall probability of occurrence. We then estimated the mean and standard deviation for any pair of clauses in the actual data using the simulated sequences.

$\boldsymbol{(X, VC_{36}, OC_{7})}$	$\boldsymbol{\rightarrow_{10} (X, VC_{57}, OC_{85})}$
storm, lash, province	... storm, cross, Cuba
quake, shake, city	... quake, hit, Iran
earthquake, jolt, city	... earthquake, hit, Iran
$\boldsymbol{(X, VC_{40}, OC_{165})}$	$\boldsymbol{\rightarrow_{10} (X, VC_{52}, OC_{152})}$
police, arrest, leader	... police, search, mosque
police, detain, leader	... police, search, mosque
police, arrest, member	... police, raid, enclave
$\boldsymbol{(SC_{39}, VC_{21}, X)}$	$\boldsymbol{\rightarrow_{10} (X, beat}$ [4]$\boldsymbol{, OC_{155})}$
sun, report, earnings	... earnings,beat,expectation
xerox, report, earnings	... earnings, beat, forecast
microsoft,release,result	... result, beat, forecast
$\boldsymbol{(X, VC_{57}, OC_{7})}$	$\boldsymbol{\rightarrow_{10} (X, cause}$ [4]$\boldsymbol{, OC_{153})}$
storm, hit, coast	... storm, cause, damage
cyclone, near, coast	... cyclone, cause, damage
earthquake,hit,northwest	... earthquake,cause,damage
quake , hit, northwest	... quake, cause, casualty
earthquake,hit,city	... earthquake,cause,damage

Table 2: Example Patterns

4.1 Pattern Detection Results

In Table 2 we present several examples of high ranking (i.e. significance) patterns with different anchorings detected by our method. The detected patterns are represented using the notation of the form $(SC_i, VC_j, X) \rightarrow_n (X, VC_{i'}, OC_{j'})$. X indicates the anchoring word. In the example notation, the anchoring word is the object of the first clause and the subject of the second (O-S for short). n indicates the maximal distance between the two clauses. The terms SC, VC or OC with a subscripted index represent the cluster containing the subject, verb or object (respectively) of the appropriate clause. For instance, in the first example in Table 2, VC_{36} indicates verb cluster no. 36, containing the verbs *lash, shake* and *jolt*, among others.

[2] Our experiments showed that increasing the distance beyond this point did not result in significant increase in the number of detected patterns.

[3] Based on Gwadera et al. (2003), dealing with a similar, though simpler, case.

[4] In two of the patterns, instead of a cluster for the verb, we have a single word - *beat* or *cause*. This is the result of an automatic post-processing stage intended to prevent over-generalization. If all the instances of the pat-

Anchoring System	Number of Patterns Found
Subject-Subject	428
Object-Object	291
Subject-Object	180
Object-Subject	178

Table 3: Numbers of patterns found ($p < 5\%$)

Table 3 lists the number of patterns found, for each anchoring system. The different anchoring systems produce quantitatively different results. Anchoring between the same categories produces more patterns than between the same noun in different grammatical roles. This is expected, since many nouns can only play a certain part in the clause (for instance, many verbs cannot have an inanimate entity as their subject).

The number of instances of patterns we found for the anchored template might be considered low, and it is likely that some patterns were missed simply because their occurrence probability was very low and not enough instances of the pattern occurred in the text. In Section 4 we stated that in this task, we were more interested in precision than in recall. In order to detect a wider range of patterns, a less restricted definition of the patterns, or a different significance indicator, should be used (see Sec. 6).

Human Evaluation: In order to better determine the quality of patterns detected by our system, and confirm that the statistical significance testing is consistent with human judgment, we performed an evaluation experiment with the help of 22 human judges. We presented each of the judges with 60 example groups, 15 for each type of anchoring. Each example group contained three clause pairs conforming to the anchoring relation. The clauses were presented in a normalized form consisting only of a subject, object and verb converted to past tense, with the addition of necessary determiners and prepositions. For example, the triplet *(police, detain, leader)* was converted to *"The police detained the leader"*. In half the cases (randomly selected), these clause pairs were actual examples (instances) of a pattern detected by our system (instances group), such as those appearing in Table 2. In the other half, we listed three clause pairs, each of which conformed to the anchoring specification listed in Section 4, but which were randomly sampled from the data, and so had no connection to one another (baseline group). We asked the judges to rate on a scale of 1-5 whether they thought the clause pairs were a good set of examples of a common relation linking the first clause in each pair to the second one.

	Instances Score	Instances StdDev	Baseline Score	Baseline StdDev
All	3.5461	0.4780	2.6341	0.4244
O-S	3.9266	0.6058	2.8761	0.5096
O-O	3.4938	0.5144	2.7464	0.6205
S-O	3.4746	0.7340	2.5758	0.6314
S-S	3.2398	0.4892	2.3584	0.5645

Table 4: Results for human evaluation

Table 4 reports the overall average scores for baseline and instances groups, and for each of the four anchoring types individually. The scores were averaged over all examples and all judges. An ANOVA showed the difference in scores between the baseline and instance groups to be significant ($p < 0.001$) in all four cases.

Achievement of Model Goals: We employed clustering in our model to overcome data-sparseness. The importance of this decision was evident in our results. For example, the second pattern shown in Table 2 appeared only four times in the text. In these instances, verb cluster 40 was represented twice by the verb *arrest* and twice by *detain*. Two appearances are within the statistical deviation of all but the rarest words, and would not have been detected as significant without the clustering effect. This means the pattern would have been overlooked, despite the strongly intuitive connection it represents. The system detected several such patterns.

The other reason for clustering was generalization. Even in cases where patterns involving single words could have been detected, it would have been impossible to unify similar patterns into generalized ones. In addition, when encountering a new clause which differs slightly from

tern in the text contained the same word in a certain position (in these examples - the verb position in the second clause), this word was placed in that position in the generalized pattern, rather than the cluster it belonged to. Since we have no evidence for the fact that other words in the cluster can fit that position, using the cluster indicator would be over-generalizing.

the ones we recognized in the original data, there would be no way to recognize it and draw the appropriate conclusions. For example, there would be no way to relate the sentence *"The typhoon approached the coast"* to the fourth example pattern, and the connection with the resulting damage would not be recognized.

5 Comparison with Previous Work

The relationship between textual features and semantics and the use of syntax as an indicator of semantics has been widespread. Following the idea proposed in Harris' Distributional Hypothesis (Harris, 1985), that words occurring in similar contexts are semantically similar, many works have used different definitions of context to identify various types of semantic similarity. Hindle (1990) uses a mutual-information based metric derived from the distribution of subject, verb and object in a large corpus to classify nouns. Pereira et al. (1993) cluster nouns according to their distribution as direct objects of verbs, using information-theoretic tools (the predecessors of the tools we use in this work). They suggest that information theoretic measures can also measure semantic relatedness.

These works focus only on relatedness of individual words and do not describe how the automatic estimation of semantic similarity can be useful in real-world tasks. In our work we demonstrate that using clusters as generalized word units helps overcome the sparseness and generalization problems typically encountered when attempting to extract high-level patterns from text, as required for many applications.

The DIRT system (Lin and Pantel, 2001) deals with inference rules, and employs the notion of paths between two nouns in a sentence's parse tree. The system extracts such path structures from text, and provides a similarity measure between two such paths by comparing the words which fill the same slots in the two paths. After extracting the paths, the system finds groups of similar paths. This approach bears several similarities to the ideas described in this paper, since our structure can be seen as a specific path in the parse tree (probably the most basic one, see Fig. 1). In our setup, similar clauses are clustered together in the same *Clustered-Clause*, which could be compared to clustering DIRT's paths using its similarity measure. Despite these similarities, there are several important differences between the two systems. Our method uses only the relationships inside the path or clause in the clustering procedure, so the similarity is based on the structure itself. Furthermore, Lin and Pantel did not create path clusters or generalized paths, so that while their method allowed them to compare phrases for similarity, there is no convenient way to identify high level contextual relationships between two nearby sentences. This is one of the significant advantages that clustering has over similarity measures - it allows a group of similar objects to be represented by a single unit.

There have been several attempts to formulate and detect relationships at a higher context level. The VerbOcean project (Chklovski and Pantel, 2004) deals with relations between verbs. It presents an automatically acquired network of such relations, similar to the WordNet framework. Though the patterns used to acquire the relations are usually parts of a single sentence, the relationships themselves can also be used to describe connections between different sentences, especially the enablement and *happens-before* relations. Since verbs are the central part of the clause, VerbOcean can be viewed as detecting relations between clauses as whole units, as well as those between individual words. As a solution to the data sparseness problem, web queries are used. Torisawa (2006) addresses a similar problem, but focuses on temporal relations, and makes use of the phenomena of Japanese coordinate sentences. Neither of these approaches attempt to create generalized relations or group verbs into clusters, though in the case of VerbOcean this could presumably be done using the *similarity* and *strength* values which are defined and detected by the system.

6 Future Work

The clustered clause model presents many directions for further research. It may be productive to extend the model further, and include other parts of the sentence, such as adjectives and adverbs. Clustering nouns by the adjectives that describe them may provide a more intuitive grouping. The addition of further elements to the structure may also allow the detection of new kinds of relations.

The news-oriented domain of the corpus we used strongly influenced our results. If we were interested in more mundane relations, involving day-to-day actions of individuals, a literary corpus would probably be more suitable.

In defining our pattern template, several elements were tailored specifically to our task. We limited ourselves to a very restricted set of patterns in order to better demonstrate the effectiveness of our model. For a more general knowledge acquisition task, several of these restrictions may be relaxed, allowing a much larger set of relations to be detected. For example, a less strict significance filter, such as the *support* and *confidence* measures commonly used in data mining, may be preferable. These can be set to different thresholds, according to the user's preference.

In our current work, in order to prevent overgeneralization, we employed a single step post-processing algorithm which detected the incorrect use of an entire cluster in place of a single word (see footnote for Table 2). This method allows only two levels of generalization - single words and whole clusters. A more appropriate way to handle generalization would be to use a hierarchical clustering algorithm. The Agglomerative Information Bottleneck (Slonim and Tishby, 1999) is an example of such an algorithm, and could be employed for this task. Use of a hierarchical method would result in several levels of clusters, representing different levels of generalization. It would be relatively easy to modify our procedure to reduce generalization to the level indicated by the pattern examples in the text, producing a more accurate description of the patterns detected.

Acknowledgments

The author acknowledges the support of EPSRC grant EP/C538447/1. The author would like to thank Naftali Tishby and Mirella Lapata for their supervision and assistance on large portions of the work presented here. I would also like to thank the anonymous reviewers and my friends and colleagues for their helpful comments.

References

Brody, Samuel. 2005. *Cluster-Based Pattern Recognition in Natural Language Text*. Master's thesis, Hebrew University, Jerusalem, Israel.

Chklovski, T. and P. Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proc. of EMNLP*. pages 33–40.

Dagan, I., O. Glickman, and B. Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Gwadera, R., M. Atallah, and W. Szpankowski. 2003. Reliable detection of episodes in event sequences. In *ICDM*.

Harris, Z. 1985. Distributional structure. *Katz, J. J. (ed.) The Philosophy of Linguistics* pages 26–47.

Hindle, Donald. 1990. Noun classification from predicate-argument structures. In *Meeting of the ACL*. pages 268–275.

Hirst, G. and D. St-Onge. 1997. Lexical chains as representation of context for the detection and correction of malapropisms. In *WordNet: An Electronic Lexical Database, ed., Christiane Fellbaum*. MIT Press.

Leacock, C. and M. Chodorow. 1997. Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database, ed., Christiane Fellbaum*. MIT Press.

Lin, Dekang. 1994. Principar - an efficient, broad-coverage, principle-based parser. In *COLING*. pages 482–488.

Lin, Dekang and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Knowledge Discovery and Data Mining*. pages 323–328.

Pedersen, T., S. Patwardhan, and J. Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proc. of AAAI-04*.

Pereira, F., N. Tishby, and L. Lee. 1993. Distributional clustering of english words. In *Meeting of the Association for Computational Linguistics*. pages 183–190.

Resnik, Philip. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*. pages 448–453.

Slonim, N., N. Friedman, and N. Tishby. 2002. Unsupervised document classification using sequential information maximization. In *Proc. of SIGIR'02*.

Slonim, N. and N. Tishby. 1999. Agglomerative information bottleneck. In *Proc. of NIPS-12*.

Torisawa, Kentaro. 2006. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of NAACL*. pages 57–64.