

Mining the Web for Reciprocal Relationships

Michael Paul, Roxana Girju, and Chen Li

Linguistics and Computer Science Departments and Beckman Institute,
University of Illinois at Urbana-Champaign

{mjpaul2, girju, chenli}@illinois.edu

Abstract

In this paper we address the problem of identifying reciprocal relationships in English. In particular we introduce an algorithm that semi-automatically discovers patterns encoding reciprocity based on a set of simple but effective pronoun templates. Using a set of most frequently occurring patterns, we extract pairs of reciprocal pattern instances by searching the web. Then we apply two unsupervised clustering procedures to form meaningful clusters of such reciprocal instances. The pattern discovery procedure yields an accuracy of 97%, while the clustering procedures indicate accuracies of 91% and 82%. Moreover, the resulting set of 10,882 reciprocal instances represent a broad-coverage resource.

1 Introduction

Reciprocity is a pervasive concept which has been studied a lot in a wide variety of fields from ethics to game theory where it is analyzed as a highly effective “tit for tat” strategy. The ethic of reciprocity (also known as the *golden rule*), for example, is a moral code born from social interaction: “*Do unto others as you would wish them do unto you*”. The golden rule appears in most religions and cultures as a standard used to resolve conflicts.

According to sociologists and philosophers, the concept of reciprocity lies at the foundation of social organization. It strengthens and maintains social relations among people, beyond the basic exchange of useful goods. Thus, the way people conceptualize reciprocity and the way it is expressed in language

play an important role in governing people’s behavior, judgments, and thus their social interactions.

In this paper we present an analysis of the concept of reciprocity as expressed in English and present a way to model it. In particular we introduce an algorithm that semi-automatically discovers patterns encoding reciprocity based on a set of simple but effective pronoun templates. We then rank the identified patterns according to a scoring function and select the most frequent ones. Using these patterns we query the web and run two unsupervised clustering procedures to form meaningful clusters of reciprocal pattern instances. The pattern discovery procedure yields an accuracy of 97%, while the clustering procedures indicate accuracies of 91% and 82%. Moreover, the resulting set of 10,882 reciprocal instances represent a broad-coverage resource.

Next we define the concept of reciprocity as expressed in English.

Reciprocity in language

The Oxford English Dictionary Online¹ defines reciprocity as “*a state or relationship in which there is mutual action, influence, giving and taking, correspondence, etc., between two parties*”, while in WordNet the verb *to reciprocate* means “*to act, feel, or give mutually or in return*”.

Reciprocity is defined as a relation between two eventualities e_o (original eventuality) and e_r (reciprocated eventuality), which can occur in various reciprocal constructions. Each eventuality is an event² or a state between two participants. Thus, the rela-

¹<http://www.oed.com/>

²We use the term “event” to denote all those actions or activities performed by people.

tion of reciprocity $\mathfrak{R}(e_o(X, Y), e_r(Z, W))$ describes a situation where the eventuality e_r is performed “in return” for e_o . Thus, reciprocity can be seen as a special type of causal relation.

The two arguments of each eventuality represent the subject and the object (direct or indirect), in this order, and they might not all be explicitly stated in the sentence, but can be inferred. Moreover, the participants of the two eventualities might or might not be the same. A few such examples are presented below with the corresponding reciprocity relations:

(1) Mary **argued with** Paul at the station.

$\mathfrak{R}(\text{argue_with}(\text{Mary}, \text{Paul}), \text{argue_with}(\text{Paul}, \text{Mary}))$ &
 $\mathfrak{R}(\text{argue_with}(\text{Paul}, \text{Mary}), \text{argue_with}(\text{Mary}, \text{Paul}))$

(2) Paul and Mary hate **each other**.

$\mathfrak{R}(\text{hate}(\text{Paul}, \text{Mary}), \text{hate}(\text{Mary}, \text{Paul}))$ &
 $\mathfrak{R}(\text{hate}(\text{Mary}, \text{Paul}), \text{hate}(\text{Paul}, \text{Mary}))$

(3) Mary likes Paul **and** he likes her, **too**.

$\mathfrak{R}(\text{like}(\text{Mary}, \text{Paul}), \text{like}(\text{Paul}, \text{Mary}))$ &
 $\mathfrak{R}(\text{like}(\text{Paul}, \text{Mary}), \text{like}(\text{Mary}, \text{Paul}))$

(4) Mary likes Paul **for** helping her sister.
 $\mathfrak{R}(\text{help}(\text{Paul}, \text{Mary's sister}), \text{like}(\text{Mary}, \text{Paul}))$ ³

As shown in the examples above, in English there are two basic types of reciprocal constructions: mono-clausal reciprocals (involving words such as *(to) hug, to agree/argue with, partner of, mutual(ly), together, each other* – examples (1) and (2)) or sentence-level reciprocals (involving two consecutive clauses – examples (3) and (4)). Most of the sentence-level reciprocals are paraphrased by coordinations or subordinations of two clauses with the same or different predicate and most of the time inverted arguments. They might also manifest various markers as shown in bold in the examples.

In this paper we focus only on sentence-level constructions when the eventualities occur in different consecutive clauses, and when the subject – object arguments of each eventuality are personal pronoun pairs which occur in reverse order in each eventuality. One such example is “**She likes him for helping her**”. Here the two eventualities are *like(he, she)* and *help(he, she)*. In this example, although the subject of the second verb is not explicitly stated, it is easily inferred. These simplifying assumptions

³We assume here that the subject of the verb *help* has been recovered and the coreference solved.

will prove very useful in the semi-supervised pattern discovery procedure to ensure the accuracy of the discovered patterns and their matched instances.

Such a resource of reciprocal event pairs can be very useful in a number of applications, ranging from question answering and textual entailment (since reciprocal event pairs encode a type of causal relation), to behavior analysis of social groups (to monitor cooperation, trustworthiness and personality), and behavior prediction in negotiations.

The paper is organized as follows. In the next section we present relevant previous work. In Section 3 we detail a semi-supervised approach of extracting patterns which encode reciprocity in English. In section 4 we extract pairs of reciprocal instances and cluster them in meaningful clusters. In section 5 we present the experimental data and results. Discussions and conclusion are presented in Section 6.

2 Previous work

Although the concept of reciprocity has been studied a lot in different disciplines such as social sciences (Gergen et al., 1980), anthropology (Sahlins, 1972), economics (Fehr and Gächter, 2000), and philosophy (Becker, 1990), linguists have started to look deeper into this problem only more recently. Moreover, to the best of our knowledge, in computational linguistics the problem is novel.

In linguistics, most of the work on reciprocity focuses on mono-clausal reciprocal constructions, in particular on the quantifiers *each other* and *one another* (Dalrymple et al., 1998; Heim, 1991; König, 2005). Most of this work has been done by language typologists (Maslova and Nedjalkov, 2005; Haspelmath, 2007) who are interested in how reciprocal constructions of these types vary from one language to another and they do this through comparative studies of large sets of world’s languages.

In computational linguistics, our pattern discovery procedure extends over previous approaches that use surface patterns as indicators of semantic relations between nouns or verbs ((Hearst, 1998; Chklovski and Pantel, 2004; Etzioni et al., 2004; Turney, 2006; Davidov and Rappoport, 2008) inter alia). We extend over these approaches in two ways: (i) our patterns indicate a new type of relation between verbs, (ii) instead of seed or hook words we

use a set of simple but effective pronoun templates which ensure the validity of the patterns extracted.

To the best of our knowledge, the rest of our reciprocity model is novel. In particular, we use a novel procedure which extracts pairs of reciprocal instances and present two novel unsupervised clustering methods which group the instance pairs in meaningful ways. We also present some interesting observations on the data thus obtained and suggest future research directions.

3 Pattern discovery procedure

Our algorithm first discovers clusters of patterns indicating reciprocity in English, and then merges the resulting clusters to identify the final set of reciprocal constructions. In this section we detail the algorithm and evaluate it in subsection 5.2.

3.1 Pronoun templates

In this paper we focus on reciprocal eventualities which occur in two consecutive clauses and have two arguments: a subject and an object. One way to do this is to fully parse each sentence of a corpus and identify coordinations or subordinations of two clauses. Then identify the subject and object arguments of each verb in each clause with the help of a PropBank-style grammatical or semantic role labeler (Kingsbury et al., 2002) and make sure they represent people named entities (as indicated by proper names, personal pronouns, etc.). Since our focus is on reciprocal constructions, we also have to keep in mind that the verbs have to have the same set of arguments (subject-object) in reverse order. Thus, noun and pronoun coreference should also be resolved at this point.

Instead of starting with such a complex and error-prone preprocessing procedure, our algorithm considers a set of pronoun templates, where personal pronouns are anchor words (they have to be matched as such). Each template consists of four personal pronouns corresponding to a subject - object pair in one clause, and a subject - object pair in the other clause. Two such examples are

“[Part1] **I** [Part2] **him** [Part3] **he** [Part4] **me** [Part5]” and

“[Part1] **they** [Part2] **us** [Part3] **we** [Part4] **them** [Part5]”,

where [Part1] - [Part5] are partitions identifying

any sequence of words. This is an elegant procedure since in English, pronouns have different cases such as nominative and accusative⁴ which identify the subject, and respectively the object of an event. This saves us the trouble of parsing a sentence to find the grammatical roles of each verb. In English, there are 30 possible arrangements of nominative - accusative case personal pronoun pairs. Thus we built 30 pronoun templates.

This approach is similar to that of seed words (e.g., (Hearst, 1998)) or hook words (e.g., (Davidov and Rappoport, 2008)) in previous work. However, in our case they are fixed and rich in grammatical information in the sense that they have to correspond to subject - object pairs in consecutive clauses.

Since the first two pronouns in each pronoun template belong to the first clause (C1), and the last two to the second clause (C2), the templates can be restated as [Part1] C1 [Part3] C2 [Part5], with the restriction that partition 3 should not contain any of the four pronouns in the template. C1 denotes “Pronoun1 [Part2] Pronoun2” and C2 denotes “Pronoun3 [Part4] Pronoun4”. Partitions 2 and 4 contain the verb phrases (and thus the eventualities) we would like to extract. For speed and memory reasons, we limit their size to no more than 5 words.

Moreover, since the two clauses are consecutive, we hypothesize that they should be very close to each other. Thus, we restrict the size of each partition 1, 3, and 5 to no more than 5 words. We then consider all possible variations of the pattern where the size of each partition varies from 0 to 5. This results in 216 possible combinations (6^3). Moreover, to ensure the accuracy of the procedure, partitions 1 and 5 should be bounded to the left and respectively to the right by punctuation marks, parentheses, or paragraph boundaries. An example of an instance matched by one such pattern is “, **I** *cooked dinner for her and she loves me for that .*”

3.2 Scoring function

One way to compute the prominence of the discovered patterns would be to consider the frequency of each of the five partitions. However, as our preliminary experiments suggest, although individual

⁴In English, the pronouns *you* has the same form in nominative and accusative.

patterns within each partition do often repeat, ranking patterns spanning all three partitions (PART1, PART3, and PART5) is problematic. Patterns with relatively long partitions (more than 2 words each) seldomly occur more than once in the entire corpus. Thus frequency would produce very little differentiation in ranking the patterns.

Thus we developed an alternative scoring system in lieu of frequencies. A sequence of size n ($seq(n)$) is an instance of a pronoun template and a subsequence of size k ($seq(k)$) is simply a substring of the sequence with $k < n$. For example, for the instance “*I love her and she loves me , too*” of length 9, there will be two subsequences of length 8: “*love her and she loves me , too*” and “*I love her and she loves me ,*”. Taking into account the frequencies of the subsequences occurring within instances of each partition, we use the following recursive scoring function (n is the length of each subsequence of size n):

$$Score(seq(n)) = \begin{cases} Disc(freq(seq(n))) + \\ \sum_{seq(n-1)} Disc(Score(seq(n-1))), & \text{if } n > 1 \\ freq(seq(n)), & \text{if } n = 1 \end{cases} \quad (1)$$

In addition, in order to ensure a valid ranking over the extracted templates with different lengths for each partition, we need to normalize the scores obtained for PART1, PART3, and PART5. In other words, we need to scale the scores obtained for each partition to discount the scores of longer partitions, so that the maximum possible score would remain the same regardless of how long the partition is. So we use the following formula to compute the discount for each of PART1, PART3, and PART5, where n is the length of the subsequence:

$$Disc(Score(seq(n))) = \begin{cases} (1.0 - fraction) * \frac{fraction^{m-n}}{m-n+1}, & \text{if } n > 1 \\ \frac{fraction^{m-n}}{m-n+1}, & \text{if } n = 1 \end{cases} \quad (2)$$

Fraction is an empirically predetermined parameter - here set to 0.5. The variable m is the length of the entire PART1, PART3, or PART5 in question.

This allows not only the frequency of the exact pattern to contribute to the score, but also occurrences of similar patterns, although to a lesser extent. And since partitions 1, 3, and 5 constitute the salient parts of the pattern as the environment for the two reciprocal clauses C1 and C2, we take the score

to be ranked as $Score(PART1) * Score(PART3) * Score(PART5)$.

We searched the 30 pronoun templates with various partition sizes on a 20 million word English corpus obtained from Project Gutenberg, the largest single collection of free electronic books (over 27,000) (<http://www.gutenberg.org>) and British National Corpus (BNC), an 100 million word collection of English from spoken and written sources. There were 2,750 instances matched which were ranked by the scoring function. There were 1,613 distinct types of patterns which generated 1,866 distinct pattern instances. Thus, we selected the top 15 patterns, after manual validation. These patterns represent 56% of the data (Table 1). All the other patterns were discarded as having very low frequencies and being very specific.

The manual validation was necessary in order to collapse some of the identified instances into more general classes. For example, the patterns “C1 and C2 to” (e.g., “*He could not hurt me and I would not wish him to.*”), “C1 and C2 in” (e.g., “*I give you and you take me in.*”), and “C1 and C2 fast said Aunt Jane” (e.g., “*He will come to her and she can hold him fast said Aunt Jane.*”) were collapsed into “C1 and C2”. This procedure can be partially solved by identifying complex verbs such as “take in”. However, we leave this improvement for future work.

Patterns	Examples
C1 [, ; :] C2	I help him; he helps me.
C1 and C2	He understands her and she understands him.
C1 and C2 [right] back	I kissed him and he kissed me back .
C1 and C2 for that	They helped us and we appreciate them for that .
C1 and C2, too	I love her and she loves me, too .
C1 when C2	He ignores her when she scolds him.
C1 whenever C2	He is there for her whenever she needs him.
C1 because C2	They tolerate us because we helped them.
C1 as much as C2	He loves her as much as she loves him.
C1 for C2 (vb-ing)	He thanked her for being patient with him.
C1 but C2	I loved her but she dumped me.
C1 for what C2	They will punish him for what he did to them.
C1 and thus C2	She rejected him and thus he killed her.
when C1, C2	When he confronted them, they arrested him.
C1 as long as C2	She will stay with him as long as he doesn't hurt her.

Table 1: The top 15 reciprocal patterns along with examples.

4 Clustering of Reciprocal Eventualities

It seems reasonable to expect that certain reciprocities could be grouped together. For example, the language used in convincing a person of something could be characterized by verbs such as $e_o = \{convince, promise, assure, beg\}$ and $e_r = \{believe, trust, choose, forgive\}$.

There are many potential uses for this sort of grouping. Having a single group label for multiple reciprocal eventuality pairs would allow us to identify certain language patterns as a particular speech act. Also, such clusters could be useful if one wants to perform a macro-level analysis of reciprocity in a specific domain. For example, examining reciprocal language could be useful in analyzing the nature of a social community or the theme of a literary work. Generalizing over many similar instances, will give us better insight into how people communicate – as reactions (effects) to other people’s actions (causes).

Thus, in this section we present a model for clustering the eventualities we extract through the process described in the previous sections. Experimental results are presented in Section 5.

4.1 Representing the data

After obtaining these patterns, we must extract pairs of eventualities of the form (e_o, e_r) . This involves both reducing the clauses into a form that is semantically representative of some eventuality, as well as determining the order of the two eventualities (i.e., if they are asymmetric).

As shown in the previous sections, each pattern contains two clauses of the form “*Pronoun_i* [Part2/4] *Pronoun_j*”, where the first pronouns is the subject and the second is the object. From each clause we extract only the non-auxiliary verb, as it carries the most meaning. We first stem the verb and then negate it if it is preceded by *not* or *n’t*. For example, “*They do not like him because he snubbed them*” is represented as the eventualities $(e_o, e_r) = (snub, \neg like)$.

Certainly, we are missing important information by excluding phrases and ignoring modality. However, these features can be difficult to capture accurately, and since inaccurate input could degrade the clustering accuracy, in this research we stick with the important and easily-obtainable features.

4.2 Ordering the eventualities

Most patterns entail a particular ordering of the two eventualities, corresponding to symmetric (e.g., “*He loves her and she loves him*”) or asymmetric eventualities (e.g., “*He ignores her when she scolds him*”). In ambiguous situations (e.g., “*He loves her and she loves him*” and “*He cheated on her and she still loves him!*”), we determine the order through clues such as the relative temporal ordering of the verbs as determined by their tense (e.g., past or present tense happens before future tense) and whether the verbs denote an action (e.g., “to chase”) or a state (e.g., “to love”). For this we rely on our previous work (Girju, 2009) where we identified the order of eventualities based on a set of such features employed in a semi-supervised model whose accuracy is 90.2%.

4.3 Modeling the relationships

The extracted eventuality pairs can be represented as a bipartite graph with a node for all e_o values in one partition, a node for all e_r values in another partition, and an edge between these nodes for each (e_o, e_r) pair. An intuitive way to cluster these eventualities is to find groups of nodes such that each node in one partition has an edge to every node in the other partition and vice versa. This is a form of hard-clustering, as membership in a cluster is strictly *yes* or *no*. The goal is that one could randomly pull an e_o and an e_r from a given cluster and the reciprocity would be valid. For example, “help” and “give” could both be reciprocated by either “thank” or “like”. Thus, given a cluster, not only is there a reciprocal relationship between verbs in the e_o group with the verbs in the e_r group, but there is often a kind of similarity relationship between the verbs within each e_o or e_r group.

This approach gives precise and concrete relations between verbs, but while it could be well-suited to some applications (such as knowledge base construction or automatic verb classification (Joanis et al., 2008)⁵) it has disadvantages in the context of grouping these verbs together. The clusters are small and sparse, and the results are difficult to interpret, as there are many overlapping clusters.

⁵These verb classes correspond to some extent to the Verb-Net (Kipper et al., 2000) or FrameNet-style (Baker et al., 1998) verb classes such as *admire, judgment*.

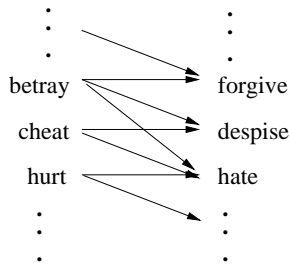


Figure 1: A sample of our data as a bipartite graph. Some edges have been omitted for readability. The nodes $\{e_o=\text{"betray"}, e_o=\text{"cheat"}, e_r=\text{"despise"}, e_r=\text{"hate"}\}$ form a cluster with our hard-clustering approach.

We instead adopt a probabilistic framework, which allows us to relax the restrictiveness of the clusters while retaining information about the strength of the pairwise relations. Thus, we design a bimodal mixture model in which we assume that each pair of eventualities (e_o, e_r) belongs to a latent class z , and each class is associated with two distinct multinomial distributions from which the two eventualities are independently drawn. Thus, the probability of generating a particular pair is:

$$P(e_o, e_r) = \sum_k^{|Z|} P(e_o|z = k)P(e_r|z = k) \quad (3)$$

Each class can be thought of as a general type of reciprocity, such as an action followed by appreciation, or an attack followed by retaliation. We should be clear that each class is characterized not by a distribution of specific pairs, but by a distribution of e_o verbs and a distribution of e_r verbs. This allows for the classification of (e_o, e_r) pairs that do not appear in the corpus. For example, if we have not seen the pair $(slap, punch)$, but we know that $(slap, hit)$ and $(kick, punch)$ belong to the same class, then it is likely that $(slap, punch)$ is in the same group.

This model can be used in a fully supervised as well as a semi-/unsupervised setting. If some or all of the class labels are unknown, we can learn the model parameters using an estimator such as Expectation-Maximization (EM) (Dempster et al., 1977). For each eventuality pair c_i in a collection C , we update $P(z = k|c_i)$ with the following equation, which represents the E-step:

$$P(z|c_i) \propto P(z)P(e_o^{(c_i)}|z)P(e_r^{(c_i)}|z) \quad (4)$$

In the M-step, we use the following update equations:

$$P(z = k) \propto \alpha + \sum_i^{|C|} P(z = k|c_i) \quad (5)$$

$$P(e_o = j|z) = \frac{\beta + \sum_i^{|C|} I(e_o^{(c_i)} = j)P(z|c_i)}{|E_o|\beta + \sum_{j'} \sum_i I(e_o^{(c_i)} = j')P(z|c_i)} \quad (6)$$

where I is a binary indicator function. The equation for $P(e_r = j|z)$ is identical to that for e_o , but with e_r instead⁶.

α and β are the hyperparameters of the uniform Dirichlet priors of $P(z)$ and $P(e_*|z)$. They can be tuned to control the level of smoothing; a value of 1.0 is equivalent to the commonly-used Laplace smoothing (Nigam et al., 2000).

4.4 Identifying polarity words

Since we are interested in analyzing how people interact, we would also like to identify the polarity (affective value) associated with each eventuality. Thus, we automatically identify polarity words in both clauses. For this we consider the standard polarity values: *Good*, *Bad*, and *Neutral*.

In the next section we present in detail the results of the evaluation.

5 Experimental data and results

5.1 Data collection

While the Gutenberg and BNC collections are useful in obtaining the frequent patterns, they do not contain a very large number of eventuality pairs to do meaningful clustering. We thus query the web through Google to easily obtain thousands of examples. We queried each of the top 15 patterns and all pronoun combinations thereof (e.g. “they * us because we * them”) and took the top 500 results for each pattern/pronoun combination $(15*30*500)$ ⁷. We then extracted the clauses from the result snippets using the procedure outlined in the previous section and ended up with 10,882 pairs

⁶We sometimes use the shorthand $P(z)$ to represent $P(z = k)$, which is updated for each particular value of z .

⁷This is because Google limits traffic. However, in the future we can acquire more instances.

(4,403 unique pairs) since some of the queries had less than 500 matched instances⁸.

5.2 Pattern discovery procedure

Since we wanted to see to what extent the 15 most frequently occurring patterns encode reciprocity, we selected a sample of 10 pattern instances matched by each pattern in the text collection obtained from the web. We presented the resulting 130 sentences (a few patterns were not frequent on the web, so we obtained a few less than 10 instances) to 2 judges who evaluated them as encoding reciprocity ('yes') or not ('no'). The judges agreed 97% of the time. Moreover, only 2.3% of the 130 pattern instances did not encode reciprocity as agreed by both judges.

These statistics show that these patterns are highly accurate indicators of reciprocity in English.

5.3 Unsupervised clustering

We can capture pattern instance clusters with no prior labeling by initializing the EM parameters randomly. In our experiments we used $\alpha = 1.0$ and $\beta = 0.01$, with varying numbers of clusters (which we denote as k). EM is sensitive to the initial parameters and can perform poorly due to many local maxima. We thus ran the algorithm several times, and saved the output with the best log-likelihood.

Results from clustering with $k = 6$ are shown in Table 2. The examples shown correspond to a random sample of 10 pairs within the top 10% of $P(e_o, e_r | cluster)$ within each cluster. We find that with larger values of k such as 30 or 50, some of the clusters become noisier, but we can capture finer-grained clusters such as $e_o = \{libel, defame\}$ and $e_r = \{sue, \neg sue\}$.

Upon a close look at the clusters in Table 2, one can see that each one seems to have a central theme. Cluster 1 seems to contain mostly positive actions reciprocated by verbs describing gratitude and appreciation. Cluster 2 has to do with cognition; Cluster 3 has to do with the way people communicate and interact. Cluster 4 captures relationships of need and desire. Cluster 5 is about love and adoration, while Cluster 6 is about hate and other negative events, and how they are reciprocated.

⁸The reciprocity dataset is available for download at <http://apfel.ai.uiuc.edu/resources.html>.

No. instances	Accuracy	
	6 clusters	9 clusters
Top 20	90.8%	82.2%
20/100	71.7%	66.1%
20/All	34.2%	26.1%

Table 3: Cluster membership accuracy for 6 and 9 clusters.

Cluster membership is defined as $argmax_c P(e_o|c) P(e_r|c)$. We took three samples of pairs: (1) the top 20 pairs with the highest $P(e_o, e_r|c)$ values, (2) a random 20 of the top 10%, and (3) a random 20 of all pairs assigned to each cluster. We presented the pairs to two judges who were asked to identify each pair as belonging to the cluster or not based on coherence; that is, all pairs labeled "yes" appear to be related in some way.

Because we fix the number of clusters, we are making the assumption that each reciprocal pair could be put into one of k groups, which is obviously an assumption that will not hold true. However, if a pair does not fit well into any of the clusters, this should be reflected by a low probability. Thus we can achieve decently high accuracy if we consider only the highest-ranked pairs. The accuracy when considering all pairs is only 34% which means that 34% of reciprocal pairs can be meaningfully placed into only 6 groups, which is actually fairly high.

A big source of inter-annotator disagreement comes from the ambiguity of certain verbs, which is a weakness of our limited representation. For example, without additional information it is not clear how a pair like (know, ask) might relate to others.

5.4 Polarity word identification

For this procedure we used the Subjectivity Clues (Wilson et al., 2005) which provides 8,220 entries. From all the 10,882 eventuality pairs, 40.1% of the total number of words were in the subjectivity lexicon, while 36.9% of the pairs had both words in the subjectivity lexicon.

Table 4 shows all possible combinations of pairs of affective values and their associated probabilities in the corpus. These values are computed for those pairs where both words have known polarity.

As one might expect, each polarity class is most likely to be reciprocated by itself: Good for Good (altruism) and Bad for Bad (retaliation). Furthermore, it is more likely that Good follows Bad ('turn

e_o	e_r	e_o	e_r	e_o	e_r	e_o	e_r	e_o	e_r	e_o	e_r
help	thank	know	respect	call	tell	need	need	love	love	hate	hate
allow	thank	trust	know	ask	give	need	trust	adore	love	attack	hate
invite	thank	tell	trust	tell	help	want	need	understand	love	attack	forgive
rescue	thank	tell	know	tell	tell	want	trust	love	adore	slap	hate
join	thank	know	know	contact	tell	want	want	teach	love	hurt	attack
inform	thank	know	trust	meet	hear	help	need	protect	love	betray	punish
join	admire	know	follow	follow	see	offer	need	feed	love	kill	hate
send	thank	give	let	watch	send	help	help	challenge	love	hit	curse
support	thank	let	like	tell	ignore	help	trust	need	love	treat	dislike
teach	owe	help	marry	confront	tell	love	need	give	love	ruin	shoot

Table 2: The clusters induced after running our unsupervised algorithm with $k = 6$ clusters. The pairs correspond to a sample of the top 10% of pairs with the highest value of $P(e_o, e_r | cluster)$ for each cluster.

	Good	Bad	Neutral	Total
Good	0.90	0.18	0.29	0.63
Bad	0.09	0.82	0.08	0.29
Neutral	0.01	0.002	0.63	0.09

Table 4: All possible combinations of pairs of affective values and their associated probabilities as found in the corpus. The numbers in the table correspond to conditional probabilities $P(row_i | col_j)$. The Total column indicates the probability of each affective class ($P(row_i)$).

the other cheek’) than that Bad follows Good.

We experimented with incorporating polarity into our clustering process. We defined 9 clusters for each combination of polarity pairs, and initialized the model by labeling the eventuality pairs where the polarity of both words was known. We then ran the EM process on all of the pairs, and since the model parameters were initialized with these 9 groups, their pairs were more likely to fit into clusters that matched their polarity. We found, however, that it had trouble clustering the less-common classes – essentially, everything but (Good, Good) and (Bad, Bad). For example, the cluster that was initialized as (Bad, Good) ended up being dominated by $e_r = thanks$ and mostly positive-polarity words as e_o . This seems to be due to the fact that many of these pairs included $e_r = thanks$ (often in sarcasm, as in “he thanked them for embarrassing him”). But there are many more words associated with *thanks* that are Good, thus those pairs were put into the same group, and the Good verbs eventually overtook the cluster. Problems such as this could perhaps be avoided with more varied labeled data.

We selected a sample of the top 20 pair instances for each of the 9 clusters of polarity pairs and gave them to 2 judges who agreed 82% of the time.

6 Discussion and Conclusions

In this paper we presented an analysis of the concept of reciprocity as expressed in English and a way to model it. The experimental results provided nice insights into the problem, but can be further improved.

We noticed that the identification of polarity words is not always enough to capture the affect of each eventuality. Thus, the text needs to be further processed to identify speech acts corresponding to each clause in the reciprocal patterns. For example, words such as “*sorry*” can be classified as negative, while the entire clause “*I am sorry*” captures the speech act of APOLOGY which is associated with good intentions. As future work, we will recluster the reciprocity pairs.

Another observation concerns the reciprocity property of *magnitude* (cf. (Jackendoff, 2005)) or *equivalence of value* between two eventualities. Most of the time reciprocal eventualities have the same or similar magnitude, as the patterns identified indicate a more or less equivalence of value – i.e., hugs for kisses, thanks for help. And most of these constructions do not focus so much on the magnitude, but on the order in which one eventuality (the effect) is a reaction to the other (the cause). However, a closer look at our data shows that there are also constructions which indicate this property more precisely. One such example is “C1 as much as C2” where even a negation in C1 or C2 might destroy the magnitude balance (e.g., “*She does not love him as much as he loves her.*”).

We would like to study this property in more detail as well. This kind of study is very important in the analysis of people’s behavior, judgments, and thus their social interactions.

References

- C. Baker, Ch. Fillmore, and J. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL 1998)*, pages 86–90, Montreal, Canada.
- L. Becker, editor. 1990. *Reciprocity*. University of Chicago Press, Chicago.
- T. Chklovski and P. Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) Conference*.
- M. Dalrymple, M. Kazanawa, Y. Kim, S. Mchombo, and S. Peters. 1998. Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy*, 21:159–210.
- D. Davidov and A. Rappoport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated sat analogy questions. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*.
- A. P. Dempster, N.M. Laird, and D. B. Rdin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38.
- O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2004. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the National Conference on Artificial Intelligence (AAAI) Conference*.
- E. Fehr and S. Gächter. 2000. Cooperation and Punishment in Public Goods Experiments. *American Economic Review*, 90:980–994.
- K. Gergen, M. Greenberg, and R. Willis, editors. 1980. *Social Exchange: Advances in Theory and Research*. New York: Plenum.
- R. Girju. 2009. Reciprocity in language. In *Technical Report*. University of Illinois at Urbana-Champaign.
- M. Haspelmath. 2007. Further remarks on reciprocal constructions. In Vladimir P. Nedjalkov, editor, *Reciprocal Constructions*, pages 2087–2115.
- M. Hearst. 1998. Automated Discovery of WordNet Relations. In Christiane Fellbaum, editor, *An Electronic Lexical Database and Some of its Applications*, pages 131–151. MIT Press, Cambridge, MA.
- I. Heim. 1991. Reciprocity and plurality. *Linguistic Inquiry*, 22:63–101.
- R. Jackendoff. 2005. The peculiar logic of value. *Journal of Cognition and Culture*, 6:375–407.
- E. Joanis, S. Stevenson, and D. James. 2008. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(3).
- P. Kingsbury, M. Palmer, and M. Marcus. 2002. Adding Semantic Annotation to the Penn Treebank. In *Proceedings of the 2nd Human Language Technology Conference (HLT 2002)*, pages 252–256, San Diego, California.
- K. Kipper, H. Trang Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 691–696, Austin, TX.
- E. König. 2005. Reciprocity in language: Cultural concepts and patterns of encoding. *Uhlenbeck Lecture*, 23.
- E. Maslova and V. Ndjalkov. 2005. Reciprocal constructions. In M. Haspelmath, M. Dryer, D. Gill, and B. Comrie, editors, *The World Atlas of Language Structures*, pages 430–433. New York: Oxford University Press.
- K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134.
- M. Sahlins, editor. 1972. *Stone Age Economics*. Chicago: Aldine-Atherton.
- P. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.
- T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology (HLT/EMNLP) Conference*.