Natural Language Engineering **17** (1): 1–36. © Cambridge University Press 2011
doi:10.1017/S1351324910000173

Modeling reciprocity in social interactions with probabilistic latent space models

ROXANA GIRJU and MICHAEL J. PAUL*
University of Illinois at Urbana-Champaign,
Urbana, IL 61801, USA
emails: girju@illinois.edu, mjpaul2@illinois.edu

(*Received 20 July 2009; revised 1 March 2010; accepted 13 May 2010*)

Abstract

Reciprocity is a pervasive concept that plays 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 a way to model it. At a larger structural level the reciprocity model will induce representations and clusters of relations between interpersonal verbs. In particular, we introduce an algorithm that semi-automatically discovers patterns encoding reciprocity based on a set of simple yet effective pronoun templates. Using the most frequently occurring patterns we queried the web and extracted 13,443 reciprocal instances, which represent a broad-coverage resource. Unsupervised clustering procedures are performed to generate meaningful semantic clusters of reciprocal instances. We also present several extensions (along with observations) to these models that incorporate meta-attributes like the verbs' affective value, identify gender differences between participants, consider the textual context of the instances, and automatically discover verbs with certain presuppositions. The pattern discovery procedure yields an accuracy of 97 per cent, while the clustering procedures – clustering with pairwise membership and clustering with transitions – indicate accuracies of 91 per cent and 64 per cent, respectively. Our affective value clustering can predict an unknown verb's affective value (positive, negative, or neutral) with 51 per cent accuracy, while it can discriminate between positive and negative values with 68 per cent accuracy. The presupposition discovery procedure yields an accuracy of 97 per cent.

1 Introduction

Reciprocity is a pervasive and important phenomenon in human life. At every level, social relationships are guided by the shared understanding that most actions call for reactions, and that inappropriate reactions require management. The ethic of reciprocity (also known as the *golden rule*), for example, is a moral code born from social interaction: '*Do onto others as you would wish them do onto you*'. The golden rule appears in most religions and cultures as a standard used to resolve conflicts.

Reciprocity has been extensively studied in a wide variety of fields from ethics to game theory, where it is analyzed as a highly effective 'tit for tat' strategy. According

*Michael J. Paul is now at John Hopkins University (mjpaul@cs.jhu.edu).

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 computational linguistics, there is extensive literature on identifying relevant information within massive data repositories and synthesizing this information into a coherent understanding of the entities and events involved.[1] In spite of this, however there exists to date no computational linguistics study of reciprocity in natural language, nor is there any model that applies this concept to the empirical study of human social interaction in unstructured data. The current state of affairs is rather surprising, given the importance of this concept in language. Illuminating the exact processes by which people interpret each other's behavior will help meet a key challenge in linguistics and computational linguistics: to help improve communication and avoid misunderstandings in sociocultural interactions.

A detailed analysis of human action and behavior (called *social dynamics*) is required for any study of social and cultural interactions. Social dynamics has been studied extensively in social networks research that combines network topology with computational models applied mostly on data from on-line networks (i.e., who talks to whom, the time and frequency of interaction – but not based on *what* is said and meant). The field of social networks theory has grown considerably in the past years as advanced computing technology has opened the door for new research. However, although this approach may describe some typical patterns, it provides limited insight into human social interactions. The reason is that these interactions most of the time are expressed through language. The limitation of current approaches to social dynamics and the spiraling amounts of online textual information generated every day require and make possible a new perspective coming, this time from the computational linguistics community.

In this paper we present an analysis of the concept of reciprocity as expressed in English and propose a series of algorithms to model it. At a larger structural level the reciprocity model will induce representations and clusters of relations between interpersonal verbs.

Our approach is bottom-up in the sense that we get new insights into the reciprocity relation based on the generalizations made from individual reciprocal relationships extracted from the web. Specifically, in this paper we introduce an algorithm that semi-automatically discovers patterns encoding reciprocity based on a set of simple yet effective pronoun templates. We then rank the identified patterns according to a scoring function and select the most frequent ones. Using these patterns we queried the web and other collections and extracted 13,443 reciprocal instances, such as the following:

(1) When he *rebuffed* her, she *sued* him.

[1] One such example is the Automatic Content Extraction (ACE) Evaluation Program (http://www.itl.nist.gov/iaui/894.01/tests/ace/).

(2) They *are criticizing* her for what she *did* to them.
(3) I *will never forgive* you for *lying* to me.

These instances represent a broad-coverage resource of reciprocal event pairs (as shown in italics in the above examples). Such a resource can be very useful in a number of applications, ranging from question answering and textual entailment (because 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.

Moreover, our database of reciprocal instances is very rich in semantic and pragmatic information, and thus can be used in various knowledge-rich applications that require reasoning and inference. For example, our unsupervised clustering procedures are performed to generate meaningful semantic clusters of reciprocal instances. We also present several extensions to these models (along with observations) that incorporate meta-attributes, such as the verbs' affective value, study gender differences between participants, consider the textual context of the instances, and automatically discover verbs with certain presuppositions.

It seems reasonable to expect that certain reciprocities could be grouped together. The clustering with transitions and affective value, for example, shows that confrontation classes, such as {*hit, attack, kill*} are more likely to be reciprocated by the *hate* class than the *forgiveness* class. 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 reciprocal relations 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).

Moreover, the gender experiments show that in social reciprocal interactions men seem to be more violent and aggressive whereas women are more forgiving. We also discover verbs that are more strongly associated with a particular gender as the initiator of an action. For example, *rape* occurs more often with men while verbs like *emasculate* are more often associated with women.

We also found that clustering of words by the textual context of the reciprocal instances yields interesting results. These show that transitions between reciprocal classes can be highly context-dependent.

Finally, we present a clustering method that automatically discovers verbs that presuppose an original eventuality – i.e., verbs such as *blame*, *forgive*, and *thank*. Such clusters can be very useful in generating inference rules for reasoning applications.

The pattern discovery procedure yields an accuracy of 97 per cent, while the basic clustering procedures indicate accuracies of 91 per cent (clustering with pairwise membership) and 64 per cent (clustering with transitions). Our affective value clustering can predict an unknown verb's affective value (positive, negative, and neutral) with 51 per cent accuracy, while it can discriminate between positive and negative values with 68 per cent accuracy. The presupposition discovery procedure yields an accuracy of 97 per cent.

These rich models of meaning will potentially pave the way toward the creation of systems with true language understanding capabilities for social interaction analysis and social inference. Moreover, the associations and clusters thus generated would hopefully draw more attention in the community to computational approaches to semantic and pragmatic analysis problems.

The paper is organized as follows. In the next section we introduce the concept of reciprocity as expressed in English and propose a formal representation for it, followed by relevant previous work in Section 3. In Section 4 we detail a semi-supervised approach of extracting patterns that encode reciprocity in English. In Section 5 we present various algorithms for extracting pairs of reciprocal instances and clustering them in meaningful clusters. In Section 6 we describe the experimental data and present the results. Discussions and conclusion are presented in Section 7.

2 Reciprocity in English

The Oxford English Dictionary Online[2] 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 (Fellbaum 1998) the verb *to reciprocate* means '*to act, feel, or give mutually or in return*'.

Following these definitions, we define reciprocity 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 event[3] or a state between two participants:

$$\Re(e_o(\mathrm{X}, \mathrm{Y}), e_r(\mathrm{W}, \mathrm{Z}))$$

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.

From a timing point of view there are two distinct possibilities:

(a) Mutual reciprocity between eventualities that occur concurrently,[4] written as $e_o(X, Y)$ & $e_r(W, Z)$, and
(b) 'in return' reciprocity, when one eventuality causes the other, written as $e_o(X, Y) \prec_c e_r(W, Z)$.

A few such examples are presented below with the corresponding reciprocity relations:

(4) Mary **argued with** Paul at the station.
argue_with (Mary, Paul) & argue_with (Paul, Mary)
(5) Paul and Mary hate **each other**.
hate (Paul, Mary) & hate (Mary, Paul)

[2] http://www.oed.com/
[3] We use the term 'event' to denote all those actions or activities performed by people.
[4] The word 'concurrently' also refers to cases like '*John and Mary chase each other*', where the action is an iterative process with mutual meaning.

(6) Mary likes Paul **and** he likes her, **too**.
like (Mary, Paul) & like (Paul, Mary)
(7) Mary likes Paul **for** helping her.
help (Paul, Mary) $\prec_c$ like (Mary, Paul)[5]

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 (4) and (5)), or sentence-level reciprocals (involving two consecutive clauses – examples (6) and (7)). Most of the sentence-level reciprocals are paraphrased by coordinations or subordinations of two clauses with the same or different predicate and inverted arguments. They might also manifest various markers or cues as shown in bold in the above 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 that occur in reverse order in each eventuality (i.e., $X = Z$ and $Y = W$). For instance, in '**She** *likes* **him** for *helping* **her**', the two eventualities are *like (she, he)* 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 will prove very useful in the semi-supervised pattern discovery procedure to ensure the accuracy of the discovered patterns and their matched instances. This procedure will be described in detail in Section 4, after a summary of relevant previous work below.

3 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 Gachter 2000), and philosophy (Becker 1990), linguists have started to look deeper into this problem only more recently.

In linguistics, most of the work on reciprocity focuses on mono-clausal reciprocal constructions, in particular on the quantifiers *each other* and *one another* (Heim 1991; Dalrymple *et al.* 1998; 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 the world's languages.

However, an in-depth study of reciprocity goes beyond the study of quantifiers, to involve issues related to semantic compositionality and pragmatic and sociocultural phenomena. One of the main goals of natural language understanding, for example, is to detect narrative events (Schank and Abelson 1977; Lehnert *et al.* 1983; Mandler 1984; Halpin and Moore 2006) and order them along the time coordinate (Chambers *et al.* 2007; Verhagen *et al.* 2007; Chambers and Jurafsky 2008; Chambers and Jurafsky 2009; Pustejovsky and Verhagen 2009). Inferring semantic relations between

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

verbs has been tackled in various ways in the literature: verb classes (Kipper *et al.* 2000; Merlo and Stevenson 2001; Joanis *et al.* 2008), selectional restrictions (Resnik 1993; Resnik and Diab 2000; Lin and Pantel 2001; Glickman and Dagan 2003; Zanzotto *et al.* 2006), and others (Hobbs *et al.* 1993; Hobbs 2005). While the literature is rich in theories (and some tools) of semantics, pragmatics, and discourse (Schank and Abelson 1977; Barwise and Perry 1985; Grosz and Sidner 1986; Levin 1993; Baker *et al.* 1998; Fellbaum 1998; Kipper *et al.* 2000; Asher and Lascarides 2003; Webber *et al.* 2003; Hovy *et al.* 2006), to our knowledge, reciprocity has not been studied in computational linguistics.

In this paper we present a pattern discovery procedure that 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, and (ii) instead of seed or hook words we use a set of simple yet 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 that extracts pairs of reciprocal instances and present various novel unsupervised clustering methods along different dimensions that group the instance pairs in meaningful ways. We also present some interesting observations on the data thus obtained and suggest future research directions.

4 Approach

In the following sections we present our approach to modeling reciprocity in English. In particular we introduce an algorithm that semi-automatically discovers patterns encoding reciprocity. We then rank the identified patterns according to a scoring function and select the first k-ranked ones. Using these patterns we queried the web and extracted 13,443 reciprocal instances that represent a broad-coverage resource.

4.1 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. We present a detailed description of the algorithm in this section and evaluate it in Section 6.

We refer to a linguistic construction discovered by our procedure as 'pattern' (a pattern type) and to an occurrence of a pattern in the corpus (a pattern token) as a 'pattern instance'.

4.1.1 Pronoun templates

In this paper we focus on reciprocal eventualities that 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. The next step would be to 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.). As 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 co-reference 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 because in English, pronouns have different cases, such as nominative and accusative,[6] 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 (Hearst 1998) or hook words (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.

As 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 '$Pronoun_1$ [Part2] $Pronoun_2$' and C2 denotes '$Pronoun_3$ [Part4] $Pronoun_4$'. 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 five 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 five 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 (i.e., 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'.

[6] In English, the pronoun *you* has the same form in nominative and accusative.

4.1.2 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 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 two words each) seldomly occur more than once in the entire corpus. Thus, frequency would produce very little differentiation in ranking the patterns. This frequency baseline procedure yielded very poor results – i.e., the top ten such patterns occurred only five to ten times in the corpus.

Consequently, we developed an alternative scoring system in lieu of frequencies. A sequence of size n (i.e., $seq(n)$) is an instance of a pronoun template and a subsequence of size k (i.e., $seq(k)$) 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 (where 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 irrespective of the length of the partition. Thus, we used 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. Moreover, as 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 thirty pronoun templates with various partition sizes on a 20-million word English corpus obtained from the Project Gutenberg, the largest single collection of free electronic books (over 27,000)[7] and the British National Corpus

[7] http://www.gutenberg.org

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

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.

(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, and 1,613 distinct types of patterns, which generated 1,866 distinct pattern instances. Thus, we selected the top fifteen patterns, after manual validation. These patterns represent 56 per cent 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.

We analyzed various pattern-ranking scores for various sizes of each partition PART1, PART3, PART5, and for all partitions. Overall, the best ranking scores are obtained for size 1 of each partition, with slight preference over size 0 or 1 for partitions 1 and 5, and size 1 for partition 3.

4.1.3 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 pronoun 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. However, these features can be difficult to capture accurately, and as inaccurate input could degrade the clustering accuracy, in this research we stick with the important and easily obtainable features. Another limitation of this representation is that the meaning of verbs, such as *tell, let*, and *want*, is not very clear in the absence of their context.

4.1.4 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'). For ambiguous reciprocal patterns (e.g., 'He *loves* her **and** she *loves* him' and 'He *cheated* on her **and** she still *loves* him!'), we rely on our recent work (Girju 2010), where we identified a set of six features employed in a semi-supervised model with an accuracy of 90.2 per cent. These are summarized next (the eventualities are referred here as e_1 and e_2, where the index represents the order in which they are mentioned in the reciprocal instance):

`F1. Reciprocal pattern`. This feature indicates one of the top fifteen patterns. Some patterns identify sequential eventualities, and thus impose an asymmetric reading. For example, 'She *married* him because he *made* her laugh' shows an asymmetric reciprocity, while the reciprocity in 'I *love* him as much as he *loves* me' is symmetric.

`F2, F3. Type of eventuality` indicates whether eventualities e_1 and e_2 are states or events. For example, verbs describing states refer to the way things 'are' – their appearance, state of being, smell, etc. (e.g., *need, hate, love*). Other verbs like *hit* and *chase* are action verbs. The values of these features are automatically determined based on an in-house list of 300 stative verbs identified from WordNet (Fellbaum 1998). The identification procedure captured the most important difference between stative and action verbs, as action verbs can be used in continuous tenses while stative verbs cannot.

This feature was borrowed from the linguistics literature on clause-level constructions, such as *each other* (König 2005). König, for example, suggests that with predicates denoting states, the relevant sentences express fully symmetric situations (e.g., 'These two *hate* each other.'), whereas event-denoting predicates are more compatible in their interpretation with a delay between the two relevant events (e.g., 'They *chase* each other.').

We hypothesize here that these observations can be extended to sentence-level reciprocal constructions between distinct verbs as well. Moreover, we show that in our dataset this delay between the two events (asymmetric instances) corresponds to 'in return' reciprocity (i.e., social causality).

`F4 and F5. Verb Modality` represents the modality of each verb (if any). Possible values are: may, would, can, shall, might, will, could, should, must.

`F6. Relative temporal order of the two eventualities`. This feature indicates if (and if yes, which) one eventuality happens before, after, or in the same time with the other eventuality. The order is simply calculated based on the tense information provided by each verb in context. For example, past simple happens before present or future tense. This feature is used to further 'disambiguate' those instances for which the symmetry property cannot be determined solely based on the pattern information (feature F1).

For example, for the pattern 'C1 and C2', if the eventualities e_1 and e_2 are states (or respectively events), then the encoded reciprocity relation is symmetric (or asymmetric, respectively). When the pattern is 'C1 as much as C2', if the eventualities e_1 and e_2 are events and the eventuality e_2 happens before e_1, then the encoded reciprocity relation is asymmetric.

In order to implement the features we first chunk parsed (Li and Roth 2001) each pattern instance and automatically identified the verbs along with their tense and modality information. Table 2 provides a summary of the feature values identified on the pattern instance corpus. For the eventuality type we use the term 'mixed' to refer to either a state or an event.

Table 2 indicates that the patterns 'C1 and C2 back', 'C1 when C2', 'C1 whenever C2', 'C1 because C2', 'C1 for C2 (vb-ing)', 'C1 for what C2', 'C1 and thus C2', 'when C1, C2', and 'C1 as long as C2' are asymmetric irrespective of the type of the two eventualities. The analysis indicates that all the other patterns can be either symmetric or asymmetric if their eventualities are either states or events, respectively. For these last patterns, when the eventuality type is mixed, the relative temporal order of the verbs identifies the order of the eventualities.

5 Modeling reciprocity

Once the reciprocal verb pairs have been extracted and the order of the eventualities identified, 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).

It would be beneficial to have an automated way of forming such clusters, because manual annotation is time-consuming with a large lexicon, and we may like to discover correlations that we do not explicitly predefine. Thus, in this section we present models for clustering the eventualities that we extracted through the process described in the previous sections. Experimental results are presented in Section 6.

Table 2. *The set of 4 (out of 6) features indicative of symmetric or asymmetric reciprocity shown here along with examples. The '$\prec_c$' symbol refers to 'in return' reciprocity.*

	Event. Type		Rel. temporal		
Patterns	e_1	e_2	**Order of event**	**Symmetry**	**Examples**
C1 [,|;|.] C2	State	State	$e_1 =_t e_2$	Symmetric	*He loves her; she loves him.*
	Event	Event	$e_1 =_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*He helped me, I helped him.*
C1 and C2	State	State	$e_1 =_t e_2$	Symmetric	*They respect him and he respects them.*
	Event	Event	$e_1 =_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*He hugs her and she elbows him.*
C1 and C2 back	State	State	$e_1 =_t e_2$	Symmetric	*She does love him and he loves her back.*
	Event	Event	$e_1 =_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*She kissed him and he kissed her back.*
C1 and C2 for that	Mixed	Mixed	$e_1 \leq_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*He destroyed her life and she hates him for that.*
C1 and C2, too	State	State	$e_1 =_t e_2$	Symmetric	*He loves her and she loves him, too.*
	Event	Event	$e_1 =_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*I chase him and he chases me, too.*
C1 when C2	Mixed	Mixed	$e_1 =_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*He ignores her when she scolds him.*
C1 whenever C2	Mixed	Mixed	$e_1 =_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*He was there for her whenever she needed him.*
C1 because C2	Mixed	Mixed	$e_1 =_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*She married him because he was good to her.*
C1 as much as C2	State	State	$e_1 =_t e_2$	Symmetric	*She enjoyed him as much as he enjoyed her.*
	Event	Event	$e_1 =_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*They hit him as much as he hit them.*
C1 for C2 (vb-ing)	Mixed	Mixed	$e_1 >_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*They thanked him for helping them.*
C1 but C2	State	State	$e_1 =_t e_2$	Symmetric	*I love her but she hates me.*
	Mixed	Mixed	$e_1 \leq_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*He tried to talk to her but she ignores him.*
C1 for what C2	Mixed	Event	$e_1 \geq_t e_2$	Asymmetric ($e_2 \prec_c e_1$)	*They will punish him* **for** *what he did to them.*
C1 and thus C2	Mixed	Event	$e_1 =_t e_2$	Asymmetric ($e_1 \prec_c e_2$)	*She rejected him and thus he killed her.*
when C1, C2	Mixed	Mixed	$e_2 \geq_t e_1$	Asymmetric ($e_1 \prec_c e_2$)	*When he started to hit them, they arrested him.*
C1 as long as C2	Mixed	Mixed	$e_2 =_t e_1$	Asymmetric ($e_2 \prec_c e_1$)	*She is staying with him as long as he is kind to her.*

Our clustering approach is such that (1) the results must be easily interpreted by human annotators; and (2) we must be able to assign cluster membership to reciprocal instances that we have not yet seen. These conditions could be satisfied under a probabilistic framework. Furthermore, such an approach is a natural way

to model cluster membership for this inherently ambiguous and context-dependent problem. Rather than having a strict and binary membership to a cluster, it would be useful to say that an instance belongs to a cluster with some likelihood or degree of membership. That is, we employ 'soft' clustering as opposed to deterministic 'hard' clustering.

Another important advantage of our probabilistic approach over traditional clustering methods is that we can easily introduce additional variables into our models to consider other interesting factors such as the participants and contexts of these reciprocal interactions.

Below we present two probabilistic models for basic verb clustering within our semantic space of reciprocal eventualities. We then introduce several extensions to these models to incorporate meta-attributes like the verbs' affective value, to model gender differences between participants, to consider the textual context of the instances, and to automatically discover verbs with certain presuppositions.

5.1 Basic model

Probabilistic generative models with hidden variables have become increasingly popular in the field of text mining and natural language processing (NLP). For example, topic models like probabilistic latent semantic analysis (pLSA) (Hofmann 1999) and latent Dirichlet allocation (LDA) (Blei *et al.* 2003) posit that each token is associated with two variables: a word, which is observable in a document, and a topic, which is unknown. Both of these model documents as a finite mixture of topics, where each topic is modeled as a multinomial distribution over words. Words with statistically strong co-occurrences are grouped into topics, and thus these models are an elegant way to cluster words based on distributional similarity.

There has also been a recent interest in using topic models for social sciences as a tool for generalizing over large amounts of data (Ramage *et al.* 2009), thus further motivating our interest in constructing latent variable models of social interactions. The data we are more interested in modeling, however, is not the raw text, but the connected verbs denoting interpersonal relationships that we have already extracted. We can represent our data as a graph where each verb is represented as a node, and each reciprocal instance (e_o, e_r) is an edge between the verbs. We are interested in capturing properties of and relations between these verbs based on their connectedness.

While it may seem limiting to model only the verb relations and not the larger text, probabilistic models of word networks have shown to be useful in NLP. For example, random walk approaches have been applied to WordNet (Fellbaum 1998) to compute the lexical relatedness of words, which is an important metric for tasks such as question answering, information retrieval, and opinion mining (Esuli and Sebastiani 2007; Hughes and Ramage 2007).

Many probabilistic models can be applied to graph and network data to cluster nodes based on connectedness. Possible applications of such models are protein interactions on relational data and social network analysis. For example, Hofman and Wiggins (2008) use a Bayesian model to discover "communities" within a

social network – that is, groups of nodes that are strongly linked to one another. *Stochastic block models* take this approach a step further modeling groups and the links between groups. However, Kemp *et al.*'s model (2006) is the most relevant to this research, as it simultaneously learns semantic concepts and the relations between them. Unfortunately, their model constrains each unit to belong to one group, which is not appropriate for our task because interactions with a verb might not be classified in the same way in all instances. For example, consider the verb *help* in the following two verb pairs: (*need–help*) and (*help–help*). In the first instance, *help* might have an altruistic meaning, but not in the second instance, where it is more likely an obligation or repayment.

Airoldi *et al.* (2008) address this shortcoming with *mixed-membership block models*, which allow nodes to belong to multiple groups, from which links to other nodes can be generated. However, we can not use this model as-is, because it assumes an undirected network, whereas our relational data can have a strict order.

We address these concerns below and describe two simple yet effective models for clustering our reciprocal verbs.

5.1.1 Clustering with pairwise membership

We propose a generative 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.

This approach is closely related to that of Parkkienn *et al.* (2009), who develop an asymmetric block model that considers memberships of the margin components of links. However, we structure our graph such that there can be multiple edges between the same two nodes, one for each (e_o, e_r) pair in our data, which allow us to consider not just the linkage between verbs but also the *frequency* of verbs, which will help us identify the eventualities that are more prevalent in a given class.

In a Bayesian model, the posterior probability is defined in terms of a prior probability that is coupled with the probabilities that can be inferred from the observed data. A natural way to define the prior probability here is with a Dirichlet(α) distribution, the conjugate prior of the multinomial distribution (Connor and Mosimann 1969). The α parameter is a vector that represents the most likely mixing proportions – that is, if an infinite number of multinomials are sampled from Dirichlet(α), the average distribution will reflect the components of α, and the variance of the distribution increases as the magnitude of α decreases.

When estimating probability distributions as a ratio of counts, 'pseudocounts' are often added to the observed counts to smooth the distributions. For example, it is a common practice to estimate language models with Laplace smoothing where a count of 1 is added to each word count, and the size of the vocabulary is added to the total number of counts. The inclusion of these pseudocounts can be derived from a Dirichlet prior where the α vector is a uniform vector and each component is 1 (Nigam *et al.* 2000). Values other than 1 can be used to increase or reduce the level of smoothing. As the magnitude of the α vector increases, the Dirichlet probability

mass becomes more concentrated at the points defined by the vector, and thus a large α value will increase the effect of smoothing.

Thus, in our generative model we assume that the multinomial distributions are first sampled from the Dirichlet distribution, after which the variable assignments are sampled from the multinomial distributions. Formally, the process by which our collection of eventuality pairs is generated has the following steps:

(1) Draw a multinomial distribution of classes θ from Dirichlet(α).
(2) Draw a multinomial distribution of e_o verbs ϕ_{oz} from Dirichlet(β) and a multinomial distribution of e_r verbs ϕ_{rz} from Dirichlet(β) for each class z.
(3) For each pair of eventualities s_i
    (a) sample a class z from θ, and
    (b) sample e_o from ϕ_{oz} and e_r from ϕ_{rz}.

Thus, the probability of generating a particular pair is

$$P(e_o, e_r) = \sum_{k}^{|Z|} P(z = k|\theta)P(e_o|z = k, \phi_o)P(e_r|z = k, \phi_r) \tag{3}$$

Pairs are thus 'clustered' together into each class z with some degree of membership. 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. It is important to note that each class is characterized not by a distribution of specific pairs but by a distribution of e_o and 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 never seen the pair (*slap*, *punch*), but we know that (*slap*, *hit*) and (*kick*, *punch*) belong to the same class, then it could be inferred that (*slap*, *punch*) belongs to the same group.

The Dirichlet priors add a layer of regularization to the model that smoothes the probability distributions. This is especially important in order to avoid overfitting to our relatively small corpus. This smoothing helps account for noise in the data and allows class assignments that have a count of zero in the corpus (i.e., to avoid zero probabilities). With these priors, however, an exact solution to the likelihood function becomes intractable, and we cannot use the popular and otherwise straightforward Expectation-Maximization (EM) algorithm (Dempster *et al.* 1977), an iterative hill-climbing procedure that will converge to a local maximum of the corpus likelihood, given an initial guess of the parameters.

A number of other inference techniques, such as variational methods (Jordan *et al.* 1998) or Markov chain Monte Carlo methods (Andrieu *et al.* 2003) can be used. In this paper we will use Gibbs sampling, a basic Markov chain Monte Carlo method, to approximate the parameters. In a Gibbs sampler, one approximately reproduces the posterior distribution by repeatedly sampling a value for each hidden variable from a distribution conditioned on the current state of the other hidden variables (Gilks *et al.* 1995).

Our basic Gibbs sampling algorithm is as follows:

- Initially assign each reciprocal pair a random class label.

Table 3. *A sample of verb classes induced when running our clustering with pairwise membership model with 12 classes. The words correspond to the 10 words with the highest value of* $P(e_o|class)$ *and* $P(e_r|class)$

Cluster 1		**Cluster 2**		**Cluster 3**	
e_o	e_r	e_o	e_r	e_o	e_r
hate	hate	trust	trust	help	thank
do	forgive	respect	respect	support	help
love	despise	followeth	forbid	understand	love
betray	love	love	thank	remember	sue
fear	fear	have	betray	tell	remember
despise	punish	find	love	let	understand
hurt	forgive	belong	belong	be	support
reject	kill	fuck	surprise	use	enjoy
leave	blame	care	find	thank	pay
abandon	leave	make	care	look	be

- For each iteration
  - for each pair of eventualities s_i in the collection C:
    - Subtract 1 from the current counts $n_{e_o=a}^{z=k}$ and $n_{e_r=b}^{z=k}$, where $s_i = (e_o = a, e_r = b)$ and the pair is currently assigned to the class $z_i = k$.
    - Sample a new class label $z_i = k^{(new)}$ from the multinomial distribution given by (4).
    - Add 1 to the counts $n_{e_o=a}^{z=k^{(new)}}$ and $n_{e_r=b}^{z=k^{(new)}}$.

The update equation is as follows:

$$P(z_i = k|e_{o,i} = a, e_{r,i} = b, \mathbf{z}_{-\mathbf{i}}, \alpha, \beta) \propto$$
$$\left(n_*^{z_i=k} + \alpha\right) \times \frac{n_{e_o=a}^{z=k} + \beta}{n_{e_o=*}^{z=k} + V_o\beta} \times \frac{n_{e_r=b}^{z=k} + \beta}{n_{e_r=*}^{z=k} + V_r\beta} \quad (4)$$

where V_o and V_r are the number of unique e_o and e_r verbs, respectively. We use the notation n_x^y to refer to the number of times that x has been assigned to y – for example, $n_{e_o=a}^{z=k}$ indicates the number of times the e_o verb a has been assigned to class $z = k$.

The conjugacy of the Dirichlet-multinomial distributions allows the hidden multinomials θ and ϕ to be marginalized out of the formula, leaving us with only the token assignments z to sample. The pseudocounts α and β are assumed to be known in the equation. We discuss the selection of these parameters in Section 6.

A sample of the clusters induced using 12 classes is shown in Table 3. These clusters show basic types of human interaction. Most of them are related to *love, hate, need* (often mutual), *desire* (often mutual), *trust* and *respect, communication, gratitude, physical affection*, and *physical attacks* – irrespective of the numbers of clusters induced.

5.1.2 Clustering with transitions

As an alternative approach, here we propose to cluster eventualities using a hidden Markov model (HMM). An HMM can model sequential data such that each piece of data is generated by some state, and the state of the next piece of the sequence depends on the current state (Rabiner and Juang 1986). This is a natural approach for our data, because e_r 'follows' e_o. (In the case of a symmetric relationship, we generated two directed links, one for each direction.)

In our case, we posit that an e_o is drawn from some verb class, and that class has some probability of being reciprocated by another class, from which e_r is drawn. We can also consider a special start/end state, which precedes the e_o class and follows the e_r class.

Thus, we use a four-node HMM, in which the nodes t_0 and t_3 belong to a designated 'start/end class' (which we will call z_0), and t_1 and t_2 belong to some classes (≥ 1) from which e_o and e_r are respectively drawn. We once again use Dirichlet priors over the distributions. The following process is used to generate the set of instances:

(1) Draw a multinomial distribution of transitions $\pi_{i,j}$ from Dirichlet(δ) for each pair of classes.
(2) Draw a multinomial distribution of verbs ϕ_z from Dirichlet(β) for each class $z \geq 1$.
(3) For each pair of eventualities s_i
   (a) sample a class i from $\pi_{0,i}$,
   (b) sample a class j from $\pi_{i,j}$,
   (c) sample e_o from ϕ_i and e_r from ϕ_j.

The Gibbs sampling update equation is:

$$P(z_{io} = j, z_{ir} = k | e_{o,i} = a, e_{r,i} = b, \mathbf{z}_{-\mathbf{i}}) \propto$$
$$(n^{z_o=j} + \gamma) \times \frac{n^{z=j}_{e_o=a} + \beta}{n^{z=j}_{e_o=*} + V_o\beta} \times \frac{n^{z_r=k} + \gamma}{n^{z_o=j} + C\gamma} \times \frac{n^{z=k}_{e_r=b} + \beta}{n^{z=k}_{e_r=*} + V_r\beta} \quad (5)$$

where C is the number of classes.

A sample of clusters induced using $C = 16$ classes is shown in Figure 1. This approach generates classes similar to those of the pairwise model introduced in the previous subsection. However, for clustering with transitions we can not assign a single class to an entire instance (e_o, e_r) – just to the verbs individually. Under this approach, we cluster *verbs* rather than reciprocal *instances*.

5.2 Affective value classes

Another interesting possibility is to group the reciprocal eventualities together based on their affective value: { *positive (Good), negative (Bad), neutral*}.

We incorporate this attribute into our HMM-based method above by associating each class z with both a distribution over verbs and a distribution over affective values. Thus, in the generative process, after choosing a class z one independently

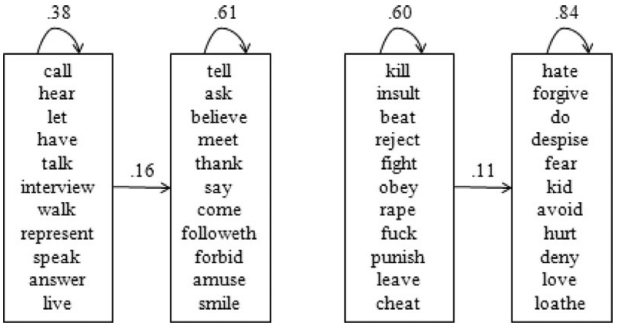

Fig. 1. A sample of verb classes induced when running our transition model with 16 classes. The words correspond to the 10 words with the highest value of $P(verb|class)$. The directed arrows correspond to the probability of transitioning from one class to the other.

samples a verb e from $P(e|z)$ and an affective value f from $P(f|z)$, which has a Dirichlet(δ) prior.

We also introduce a parameter λ that determines the probability that f is actually drawn from $P(f|z)$, while $(1-\lambda)$ would be the probability that f is chosen at random. This helps account for noise in our subjectivity lexicon. For this procedure we used the Subjectivity Clues lexicon (Wilson *et al.* 2005), which provides 8,220 entries, where content words are labeled with their corresponding affective value.

Thus, the probability of generating an affective value and an eventuality (f, e) is

$$P(f,e) = \sum_{k}^{|Z|} P(z=k|z_{prev}) \left(\lambda P(f|z=k) + (1-\lambda)\frac{1}{F}\right) P(e|z=k) \tag{6}$$

where F is the number of possible affective values (in our case, 3).

The Gibbs sampling update equation is

$$\begin{aligned} P(z_{io}=j, z_{ir}=k | e_{o,i}=a, e_{r,i}=b, f_{o,i}=c, f_{r,i}=d, \mathbf{z_{-i}}, \alpha, \beta) \propto \\ (n^{z_o=j}+\gamma) \times \frac{n^{z=j}_{e_o=a}+\beta}{n^{z=j}_{e_o=*}+V_o\beta} \times \frac{n^{z_r=k}+\gamma}{n^{z_o=j}+C\gamma} \times \frac{n^{z=k}_{e_r=b}+\beta}{n^{z=k}_{e_r=*}+V_r\beta} \\ \times \left((1-\lambda)\left(\frac{1}{F}\right) + \lambda\frac{n^{z=j}_{f=c}+\beta}{n^{z=j}_{f=*}+F\delta}\right) \times \left((1-\lambda)\left(\frac{1}{F}\right) + \lambda\frac{n^{z=k}_{f=d}+\beta}{n^{z=k}_{f=*}+F\delta}\right) \end{aligned} \tag{7}$$

In the case that a verb's affective value is not known, we simply ignore this component and compute $P(z|e_o, e_r)$ as we do in the previous section. In other words, $\lambda = 0$ in these cases.

Aside from empirical analyses, a potential use for this model is to predict the affective value of words whose value is unknown. Once the model is learned, we can compute $P(f|word)$ by marginalizing across all classes. That is,

$$P(f|word) = \sum_{z}^{C} P(f|z)P(z|word) \tag{8}$$

where $P(z|word) = \frac{P(word|z)P(z)}{P(word)}$ by Bayes theorem (Mitzenmacher and Upfal 2005).

Figure 2 shows a sample of clusters induced under this model with twelve classes along with their most probable affective value and some of the transitions between them. It is important to point out that the distribution over affective values is computed only from the verbs where this value is in the subjectivity lexicon.

Figure 2 shows that the class on the far left is positive with *forgive* as the top verb and the class on the far right is negative with *hate* as the top verb. In what concerns the two classes in the middle, both have to do with (usually physical) confrontation, but the one on the left (with {*ignore, slap, dump*}) seems to be milder – so, it is more likely to be reciprocated by the *forgiveness* class than the *hate* class. Conversely, the confrontation class on the right (with {*hit, attack, kill*}) is more likely to be reciprocated by the hate class than the forgiveness class.

5.3 Gender differences

The pairwise clustering method proposed above can also be extended to model gender differences – specifically, one would like to compare how men and women reciprocate.

We have previously explained how topic models like pLSA and LDA can be used to cluster words by distributional similarity, which relates to our verb clustering. The cross-collection mixture model (Zhai *et al.* 2004) and cross-collection LDA (ccLDA) (Paul and Girju 2009) extend these topic models to be applied across multiple collections of text, by allowing each topic to be associated with a global language model as well as a model for each collection. Thus, within each topic one can learn what is common to all collections as well as what is unique to each collection.

In previous work we showed that ccLDA can be very useful in discovering different perspectives and cultural differences in text collections (Paul and Girju 2009). If we apply this idea to our clustering of reciprocal verbs, we can compare and contrast the verbs associated with each gender and see if one gender is more likely to reciprocate in certain ways. In our model, each gender is analogous to a text collection, and within each cluster there are verbs that are unique to each gender as well as those that are common to both.

Under this model, each reciprocal pair is assigned a class z as well as a binary variable x, which denotes whether e_r was drawn from the gender-dependent or gender-independent distribution. The idea behind this model is that a reciprocal instance is generated by first choosing a class z from $P(z)$, then choosing a verb e_o from $P(e_o|z)$. Then we choose a value (0 or 1) for x from $P(x|z)$, which determines whether the e_r verb is chosen from the gender-neutral or gender-specific distribution. If $x = 0$, then we choose e_r from $P(e_r|z, x = 0)$, otherwise if $x = 1$, we choose this from the gender-specific distribution according to $P(e_r|z, x = 1, gender)$.

The complete generative process is as follows:

(1) Draw a multinomial distribution of classes θ from Dirichlet(α).
(2) Draw a multinomial distribution of e_o verbs ϕ_{oz} from Dirichlet(β).
(3) Draw a gender-independent multinomial distribution of e_r verbs ϕ_{rz} from Dirichlet(β) for each class z.

Table 4. *A sample of reciprocal classes induced when running our gender differences model with 12 classes. The words correspond to the 10 words with the highest value of* $P(e_o|class)$, $P(e_r|x=0,class)$, $P(e_r|x=1,class,male)$, *and* $P(e_r|x=1,class,female)$. *The verbs in the gender-specific distributions represent ways in which the respective genders are more likely to reciprocate than the other in this class*

e_o	e_r – **Both**	e_r – **Male**	e_r – **Female**
tell	respect	bend	forgive
respect	believe	beat	reject
embarrass	bang	attack	maintain
greet	divorce	divorce	hear
cheat	reject	revile	believe
avoid	beat	rape	respect
stop	greet	resent	advise
slap	remember	strangle	awe
nag	bend	manipulate	¬ shop
remember	kill	tell	¬ hit

(4) Draw a gender-dependent multinomial distribution of e_r verbs σ_{zg} from Dirichlet(δ) for each class z and each gender g.
(5) Draw a binomial distribution ψ_z from Beta(γ_0, γ_1) for each class z.
(6) For each pair of eventualities s_i
   (a) choose a gender g,
   (b) sample a class z from θ,
   (c) sample x from ψ_z,
   (d) sample e_o from ϕ_{oz},
   (e) If $x = 0$, sample e_r from ϕ_{rz};
   else if $x = 1$, sample e_r from σ_{zg}.

The Gibbs sampling update equation is

$$P(z_i = k, x_i = 0 | e_{o,i} = a, e_{r,i} = b, \mathbf{z_{-i}}, \alpha, \beta, \gamma) \propto$$

$$(n_*^{z_i=k} + \alpha) \times \frac{n_{x=0}^{z=k} + \gamma_0}{n_*^{z=k} + \gamma_0 + \gamma_1} \times \frac{n_{e_o=a}^{z=k} + \beta}{n_{e_o=*}^{z=k} + V_o\beta} \times \frac{n_{e_r=b}^{z=k,x=0} + \beta}{n_{e_r=*}^{z=k,x=0} + V_r\beta} \tag{9}$$

$$P(z_i = k, x_i = 1 | g_i = j, e_{o,i} = a, e_{r,i} = b, \mathbf{z_{-i}}, \alpha, \delta, \gamma) \propto$$

$$(n_*^{z_i=k} + \alpha) \times \frac{n_{x=1}^{z=k} + \gamma_1}{n_*^{z=k} + \gamma_0 + \gamma_1} \times \frac{n_{e_o=a}^{z=k} + \beta}{n_{e_o=*}^{z=k} + V_o\beta} \times \frac{n_{e_r=b}^{z=k,x=1,g=j} + \delta}{n_{e_r=*}^{z=k,x=0,g=j} + V_r\delta} \tag{10}$$

It is important to note that we use a Beta prior for $P(x)$, which is simply the bivariate analog of the Dirichlet distribution.

To determine the gender, we considered the reciprocal instances where the subject of the e_r eventuality is *he* or *she*. Table 4 shows a sample of these results with modeling with twelve classes.

In general, it seems that men are more violent and aggressive, whereas women are more forgiving. This depends on the reciprocal class, though. Consider the cluster

whose e_o words include *punish*, *refuse*, *criticize*, and *reject*. The top e_r words for men are *accept*, *hug*, *tolerate*, and *owe*. On the other hand, the top e_r words for women include *cheat*, *dump*, and *despise*. It seems that men are more forgiving in the face of criticism and rejection, while women are more forgiving in response to cheating and embarrassment. Furthermore, it seems that men and women are generally mutually respectful; it is only when that respect is broken that their responses may differ.

Some verbs are more strongly associated with men (e.g., *rape*), and some verbs that are more common to men, such as *hire* and *arrest*, are likely due to the prominence of men in authoritative positions. The verb *emasculate* was common in the female distributions (observation supported by its definition – cf. WordNet). Other words that were frequently associated with women were *nag* and *idolize*.

5.4 Considering context

So far we have focused only on the modeling of the graph data (i.e., our network of verbs linked by edges), and so the clustering approaches presented in the previous subsections do not consider the context in which the verbs appear. A model that takes into consideration the verbs as well as the words *surrounding* them would give us new insights into the contexts under which certain types of reciprocal interactions arise.

It is possible to combine the power of block models, which cluster nodes in graphs, with topic models, which cluster words that appear in text. For example, in topic modeling with network regularization (Mei *et al.* 2008), topic mixtures of documents are assumed to be similar to the documents they are connected to in a network such as the web. Relational topic models (Chang and Blei 2009) model both the text of documents and the links between them.

We will use these ideas to construct a model of the reciprocal verb network as well as of the words associated with each reciprocal pair. The joint modeling of these components will help us to give more meaning to the reciprocal classes thus induced.

We propose a model based on the clustering with transitions model that includes both the reciprocal eventualities e_o and e_r as well as their *context window*, defined as the W words before the reciprocal pattern, the W words after the pattern, and any words within the pattern itself, excluding the pronouns and the verbs e_o and e_r. We model the instances and their context window such that the words in the context window are dependent on some context c. The verb class z_o of e_o depends on c and the class z_r of e_r depends on both c and z_o. Furthermore, we say that some words in the context window can come from some 'background' word distribution that is independent of the context. This allows us to account for common words that do not fit into any context, such as *the* and *what*.

To generate a reciprocal instance and a context window under this model, a context c is first chosen according to $P(c)$. The reciprocal classes are sampled from $P(z = i|c, z_{prev} = 0)$ and $P(z = j|c, z_{prev} = i)$, and the eventualities are sampled from $P(e_o|z = i)$ and $P(e_r|z = j)$. Finally, the words of the context window are generated independently. Each word is associated with a binary variable x, as was done in our

gender differences model, which is sampled from $P(x)$. If $x = 0$, then the word w comes from the background distribution $P(w|B)$, otherwise it is sampled from the context's word distribution $P(w|c)$.

The complete generative process is as follows:

(1) Draw a multinomial distribution of contexts θ_c from Dirichlet(α) for each context c.
(2) Draw a multinomial distribution of transitions $\pi_{i,j,c}$ from Dirichlet(δ) for each pair of classes and each context c.
(3) Draw a multinomial distribution of verbs ϕ_z from Dirichlet(β) for each class $z \geq 1$.
(4) Draw a multinomial distribution of verbs σ_c from Dirichlet(β) for each class c and for the background model.
(5) Draw a Bernoulli distribution ψ from Beta(γ_0, γ_1).
(6) For each instance
  (a) sample a context c from θ,
  (b) sample a class i from $\pi_{0,i,c}$,
  (c) sample a class j from $\pi_{i,j,c}$,
  (d) sample e_o from ϕ_i and e_r from ϕ_j.
  (e) For each word w_k in the context window
    (i) sample x from ψ, and
    (ii) if $x = 0$, sample w_k from σ_B;
    else if $x = 1$, sample w_k from σ_c.

During Gibbs sampling, for each instance s_i we sample a context c_i, verb classes z_{io} and z_{ir}, and assignments of x_j for each word in the context window using the following equations:

$$P(c_i = m|\mathbf{x_i}, z_{io} = a, z_{ir} = b, \mathbf{c_{-i}}, \alpha, \beta, \gamma) \propto$$

$$(n^{c=m} + \alpha)\frac{n^{z_o=a} + \delta}{n^{c=m} + C\delta} \times \frac{n^{z_r=b} + \delta}{n^{z_o=a,c=m} + C\delta} \times \prod_{w_j \in a_i | x_j = 1} \frac{n^{c=m}_{w_j} + \beta}{n^{c=m}_{*} + V_c\beta} \tag{11}$$

$$P(z_{io} = j, z_{ir} = k|c_i = m, e_{o,i} = a, e_{r,i} = b, \mathbf{z_{-i}}) \propto$$

$$\frac{n^{z_o=j} + \delta}{n^{c=m} + C\delta} \times \frac{n^{z=j}_{e_o=a} + \beta}{n^{z=j}_{e_o=*} + V_e\beta} \times \frac{n^{z_r=k} + \delta}{n^{z_o=j,c=m} + C\delta} \times \frac{n^{z=k}_{e_r=b} + \beta}{n^{z=k}_{e_r=*} + V_e\beta} \tag{12}$$

$$P(x_j = 0|\mathbf{s_i}, c_i = m, \mathbf{w_{-i}}) \propto \frac{n^{s=i}_{x=0} + \gamma_0}{n^{s=i}_{x=*} + \gamma_0 + \gamma_1} \times \frac{n^{c=B}_{w_j} + \beta}{n^{c=B}_{w_j} + V_c\beta} \tag{13}$$

$$P(x_j = 1|\mathbf{s_i}, c_i = m, \mathbf{w_{-i}}) \propto \frac{n^{s=i}_{x=1} + \gamma_1}{n^{s=i}_{x=*} + \gamma_0 + \gamma_1} \times \frac{n^{c=m}_{w_j} + \beta}{n^{c=m}_{w_j} + V_c\beta} \tag{14}$$

V_e is the number of unique eventualities and V_c is the size of the vocabulary of the context windows.

Table 5 shows a sample of contexts induced when running our model with twenty verb classes, thirty contexts, and a context window of size $W = 10$. We also show

Table 5. *The top words in different context clusters as well as the 'background' model for words that appear independently of any context. The words correspond to the top value of* $P(word|context)$

Background	**Context 1**	**Context 2**	**Context 3**
said	past	hear	greeted
love	remember	sheep	dimly
know	memories	follow	discharge
loved	smiling	voice	quiet
don	doubt	jesus	bedroom
first	sped	sea	worshiped
say	vain	unto	delight
man	showed	worthy	women
tell	hair	satisfaction	pleasure
think	forgotten	thy	amused
mr	cheeks	christ	passionately
father	looking	eternal	ripened

the top words for the background model, which represents words that are likely to appear independently of the context.

The words in each of the context clusters are fairly related. For example, Context 1 is clearly about memory, with words such as *past*, *remember*, and *memories*. Furthermore, with words, such as *hair*, *cheeks*, and *looking*, we might infer that it is specifically about remembering the way a person looks. As one might expect, this context is most likely to transition to the verb class characterized by words such as *love*, *kiss*, *adore*, and *honor*.

Context 2 is clearly biblical, with the presence of words like *Jesus* and *Christ*, as well as older English words like *thy* and *thou*. This context is most likely to generate an e_o from the class characterized by *know*, *admire*, and *obey*, and it is most likely to then reciprocate with the class, including verbs such as *follow*, ¬ *blame*, and ¬ *want*.

Context 3 is most likely to transition to the *love* and *adoration* class, which seems sensible, with words like *pleasure* and *passionately*.

5.5 Presuppositions

Some of the identified reciprocal eventualities are presupposition-rich verbs – i.e., they presuppose the existence of an original eventuality for which they are performed 'in return'. Identifying automatically such verbs would be useful for semantic inference and behavior prediction.

Our dataset shows that some verbs, such as *thank* and *forgive*, necessarily presuppose an original eventuality e_o. Some verbs like *hate* strongly presuppose an e_o – unlike *love*, which most of the time is unconditional, one usually only hates someone for something they did. As it may not be possible to distinguish if a verb has this property *necessarily* using only our distributional approaches, we tried instead to identify verbs that have this property to a reasonable degree.

Table 6. *The results of our presupposition discovery procedure and the presupposition classes induced. The words correspond to the 10 words with the highest value of* $P(e_o|class)$ *and* $P(e_r|class)$*, and the* e_r *words can be said to presuppose an* e_o

Good–Good		**Bad–Good**		**Bad–Bad**	
e_o	e_r	e_o	e_r	e_o	e_r
help	thank	do	forgive	do	hate
support	appreciate	trouble	pardon	hurt	blame
trust	honor	disturb	excuse	betray	punish
do	congratulate	interrupt	thank	reject	kill
thank	bless	insult	console	torture	arrest
rescue	praise	deceive	praise	insult	reproach
bless		abandon	reward	mislead	chastise
spare		forget	admire	abandon	resent
enlighten		betray		distract	rebuke
pardon		disappoint		punish	berate
enable		eliminate		criticize	despise

A quick analysis of our dataset show that e_r seems to be more likely to have this presupposition property in the *for what* and *for vb-ing* patterns than the others. Following this hypothesis, these verbs were clustered such that we separate instances that are more likely to appear in these two patterns than the others. This was done with the basic pairwise clustering method under some constraints.

The clusters were initialized so that the instances from the *for what* and *for vb-ing* patterns are placed into four clusters depending on the affective value of the verbs: Good–Good, Bad–Bad, Bad–Good, and Good–Bad. Everything else was placed into cluster 0. Everything that was initially placed into cluster 0 must remain there, but the instances in clusters 1–4 can move between either their initial cluster or cluster 0. Thus, instances that are more representative of the *for what* and *for vb-ing* patterns will end up in clusters 1–4, while instances that are more like the rest of the corpus will be separated out.

Table 6 shows the three significant clusters (the Good–Bad class did not yield anything) induced by this procedure.

The e_r verbs can be said to have a strong presupposition property – Table 6 shows ten words with the highest probability in that class, although two of the classes had fewer than ten words that were ever assigned to them during sampling.

6 Experimental data and results

6.1 Data collection

While the Gutenberg and BNC collections are useful in obtaining the frequent patterns, they do not contain a very large number of reciprocal eventuality pairs to do meaningful clustering. We thus queried the web through Google to easily obtain thousands of examples. We queried each of the top fifteen patterns and all pronoun

combinations thereof (e.g. '*they * us because we * them*') and took the top 1,000 results for each pattern/pronoun combination (15*30*1000).[8] We then extracted the clauses from the result snippets using the procedure outlined in the previous section and obtained 10,822 pairs.[9]

To increase coverage, we also extracted these patterns from the Gutenberg corpus, giving us an additional 2,561 instances, for a total of 13,443 instances (5,162 unique instances).

We performed part-of-speech tagging (Tsuruoka and Tsujii 2005) and lemmatization (Minnen *et al.* 2000) of the text before extracting the reciprocal verb pairs. Our data contain 1,608 unique verbs.

6.2 Pattern discovery procedure

Since we wanted to see to what extent the fifteen most frequently occurring patterns encode reciprocity, we selected a sample of ten 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 ten instances) to two judges who evaluated them as encoding reciprocity ('yes') or not ('no'). The judges agreed 97 per cent of the time. Moreover, only 2.3 per cent 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.

6.3 Unsupervised clustering

The Gibbs sampler should be run for some number of iterations until the distribution is stationary – called the *burn-in period* – and then a number of samples has to be collected and averaged. Some number of iterations (called the *lag*) should pass between sample collection (Heinrich 2008). Unless otherwise specified, in our experiments we ran our Gibbs samplers for 500 iterations, with a 300-iteration burn-in period and a 20-iteration lag.

The selection of appropriate values for the parameters, such as the number of clusters and the Dirichlet parameters (e.g., α, β), is done qualitatively. While there are ways to automatically learn these parameters to optimize the data likelihood (Wallach 2006; Li *et al.* 2007), it has been observed that increased likelihood does not always produce better clusters in terms of semantic coherence (Chang *et al.* 2009), which is what is important in this research. We thus use a 'trial and error' approach to setting the parameters. As a starting point, we use the observation that 0.01 tends to be a good Dirichlet parameter for distributions over words, and the Dirichlet parameter for distributions over classes tends to be higher, around the order of 1.0 (Griffiths and Steyvers 2004). We then adjust these parameters as necessary until

[8] This is because Google limits traffic. However, more instances can be acquired in the future.
[9] The reciprocity dataset is available for download at our group's webpage (*Semantic Frontiers*): http://apfel.ai.uiuc.edu/resources.html

the results are meaningful, as is often done in this type of unsupervised research (Chang *et al.* 2009). Our evaluations rely on human judgments and it is not feasible to have the judges evaluate results for every combination of parameters. Thus, we selected the final parameters that yielded reasonable-looking clusters before passing the results to the judges.

As for choosing the number of classes, we find that the results are very noisy when using a large number of classes (e.g., fifty), and the clusters tend to be reasonably coherent in the ten–twenty range. Qualitatively, we find that the same kinds of clusters are induced with small variations of this number (e.g., ten vs. fifteen), so we simply used varying values within this range in the different experiments.

6.3.1 Pairwise clustering

A sample of the clusters induced using twelve classes with the parameters $\alpha = 1.0$ and $\beta = 0.01$ is shown in Table 3, which has been introduced in Section 5.

Cluster membership is defined as $argmax_c\ P(e_o|c)\ P(e_r|c)$. We presented the top nineteen pairs of each cluster 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. The judges agreed on 199 pairs out of which 182 were correct and 17 were incorrect (with a Kappa coefficient (Cohen 1960) of 0.50).

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 (*let, do*) might relate to other pairs.

6.3.2 Clustering with transitions

A sample of these clusters induced using $C = 16$ classes with parameters $\delta = 1.0$ and $\beta = 0.01$ is shown in Figure 1 introduced in Section 5.

To evaluate how well this approach clusters verbs together, we presented the results (the top ten words in each cluster) of modeling with sixteen classes to two judges. Of the sixteen clusters, the judges agreed that seven clusters were coherent and four were incoherent.

To compare against a state-of-the-art baseline, we clustered our verb network via spectral clustering, a popular graph-based clustering approach. We induced sixteen clusters by minimizing the normalized graph cut, using the Graclus software[10] (Dhillon *et al.* 2007). Each verb was assigned to exactly one cluster. We then ranked the words in each cluster in descending order of frequency and showed the top ten words to two judges. Of the sixteen clusters, the judges agreed that three were coherent and nine were incoherent (with a Kappa score of 0.46).

This approach did induce some verb clusters that were found in the results of our own models, such as classes representing *hate*, *affection*, and *physical attacks*.

[10] http://www.cs.utexas.edu/users/dml/Software/graclus.html

Table 7. *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* ($P(row_i)$)

	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

However, it found many clusters that lacked any semantic coherence, because of strong-but-noisy connectedness between low-frequency verbs.

It is also important to note that any graph clustering baseline we might use would not capture the reciprocal relations between verb clusters. Whereas our HMM-based approach learns a matrix of transitions between classes, there is no information in the spectral clustering output about which classes are likely to reciprocate other classes (and indeed, this method assumes that the graph is undirected). While a basic graph clustering algorithm might induce verb clusters, it would not model reciprocity as we do in this research.

6.3.3 Clustering with affective value

Overall, 31.4 per cent of the verbs in our corpus were found in the subjectivity lexicon, and 22.0 per cent of our reciprocal pairs had both words in the lexicon.

Table 7 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 and Bad for Bad (retaliation). Furthermore, it is more likely that Good follows Bad ('turn the other cheek') than that Bad follows Good.

Figure 2, introduced in Section 5, shows a sample of classes induced under this model with twelve classes with parameters $\alpha = 1.0$ and $\beta = \delta = 0.01$ along with their most probable affective values and some of the transitions between them. We set $\lambda = 0.8$. It is important to point out that the distribution of affective values is computed only from the verbs where this value is in the subjectivity lexicon.

One way to evaluate this clustering method is to test its ability to predict the affective value of the unknown words, as described earlier. Thus, we learn the model with twenty classes and $\lambda = 0.8$, then compute $P(f|word)$ for words that were not in the subjectivity lexicon. For each affective class, we took the thirty words with the highest value of $P(f_i|word)$, although some affective classes had fewer than thirty words assigned to them.

The remaining seventy-five words were presented (in a random order) to two judges to rate as *positive* (Good), *negative* (Bad), or *neutral*. The judges agreed on fifty-one of the words (with a Kappa score of 0.49). The low agreement is mostly

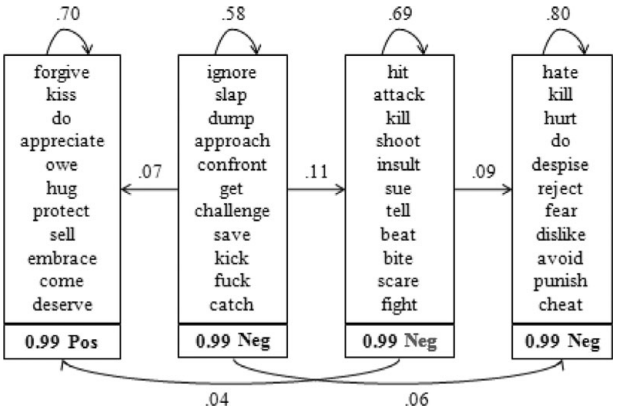

Fig. 2. A sample of verb classes induced when running our affective class model with 12 classes. The words correspond to the 10 words with the highest value of $P(verb|class)$. The directed arrows correspond to the probability of transitioning from one class to the other. At the bottom of each class is its most probable affective value, $P(aff|class)$.

due to disagreement over whether words are neutral or polar – for example, a word like *recognize* could be seen as having a positive connotation, or it could be considered neutral. In some situations, however, generic verbs, such as *do* and *come*, have a positive connotation in reciprocal contexts – hence it is not completely wrong to assume positive value for these verbs, as far as they are used in reciprocity constructions with other verbs from a certain class. However, this issue of context-dependent or context-independent affective values of verbs should be further studied.

The affective value of twenty-six words out of the total of fifty-one were correctly identified by our procedure, while twenty-five were incorrect. However, seventeen of these twenty-five incorrect words should have been *neutral*, which suggests that this approach mainly fails at discriminating positive and neutral or negative and neutral, rather than positive and negative. Of the twenty-five verbs that are positive or negative, it correctly classified the seventeen verbs.

6.3.4 Clustering gender differences

For these experiments, we could consider interactions between men/women and any other person, but we find that it is more interesting to see how men and women interact with each other. Thus, we consider instances where each participant is male and female (e.g., '*he * her because she * him*'). Table 4 shows a sample of these results with modeling with twelve classes with parameters $\alpha = 1.0$, $\beta = \delta = 0.01$, and $\gamma_0 = \gamma_1 = 1.0$. This Table was introduced in Section 5 where we also detailed our observations obtained from these experiments.

6.3.5 Clustering with context

When experimenting with this model, we ran the Gibbs sampler for 1,200 iterations, with a burn-in period of 800 iterations and a fifty-iteration lag.

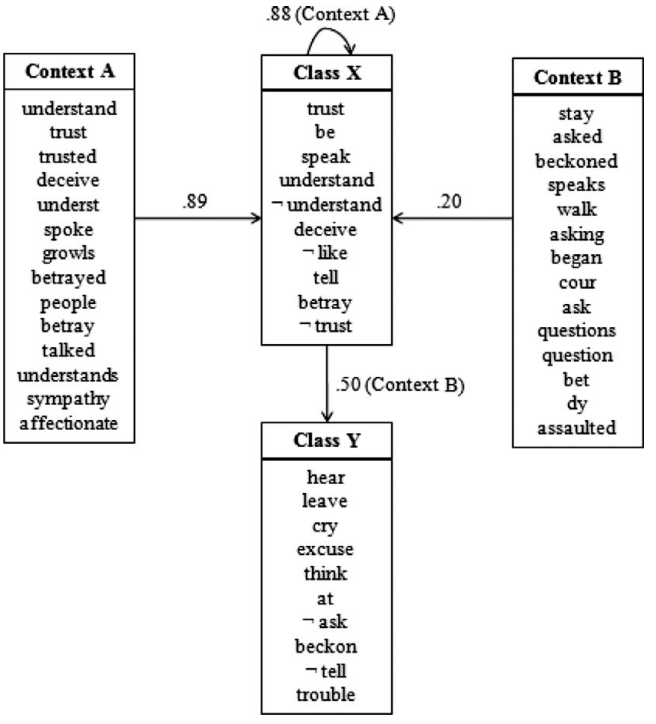

Fig. 3. An example of a word distribution of two contexts and their transition to verb classes. The words correspond to the 10 words with the highest value of $P(word|context)$ and $P(verb|class)$. The transitions correspond to $P(class_o|context)$ and $P(class_r|class_o, context)$.

Table 5, introduced in Section 5, shows a sample of contexts induced when running our model with twenty verb classes, thirty contexts, and a context window of size $W = 10$ (with $\alpha = 1.0, \beta = \sigma = 0.01, \delta = 0.1, \gamma_0 = \gamma_1 = 1.0$). We also show the top words for the background model, which represents words that are likely to appear independently of the context.

The way a verb class is reciprocated can depend on the context, as seen in Figure 3. The Figure shows two different contexts – Context A is about trust and understanding; Context B seems to be about asking questions and perhaps requesting permission. Two reciprocal verb classes, X and Y, are shown in the figure. Both contexts are most likely to generate an e_o from Class X, which is also about trust and understanding. Given Context A, Class X will most likely be reciprocated by itself. However, given Context B, Class X will most likely be reciprocated by Class Y, which seems to indicate a communicative response.

We also experimented with a window size $W = 50$, which did produce a few similar contexts, such as the biblical one. Most of the context clusters become much noiser, perhaps because it is harder to find long-range relatedness with such a small corpus size.

In general, this procedure struggles due to our corpus size – most of the Google snippets contain only the patterns themselves, and thus the context window for those instances consists of only the words (except for e_o, e_r, and the pronouns) within the

short patterns, which is not enough. This leaves us with less than 3,000 instances that have a complete context window, and because these come from such varied sources, our data is contextually very sparse. Thus, our approach fails to find many consistent and coherent clusters of words for these contexts.

However, we offer these examples as evidence that our model is capable of discovering such clusters, provided there is enough data to statistically discover strong correlations.

6.4 Discovering presuppositions

Table 6, already introduced in Section 5, shows the three significant clusters (the Good–Bad class did not yield anything) induced by this procedure (with $\alpha = 1000.0$ and $\beta = 0.01$). The large α value is used to help even out $P(z)$, otherwise $P(z = 0)$ is so large that few verbs make their way into clusters 1–3.

To judge the effectiveness of this approach, we presented the verbs identified through this process as having presuppositions (thirty-five unique verbs) to two judges, who were asked to rate each verb as 'yes – this verb has presuppositions to some degree', or 'no – this verb does not presuppose anything'. The judges disagreed on four of the thirty-five verbs. Moreover, when considering the remaining thirty-one verbs, the judges agreed that one verb did not presuppose anything and the other thirty do have presuppositions.

Indeed, most of the verbs identified by this approach have presuppositions, although certainly the recall is not perfect. For example, a verb such as *retaliate* clearly has this property, but it only appears once in our entire corpus, and thus cannot be distinguished from noise – the sparsity of our dataset is a limiting factor in this regard. Nonetheless, this seems to be a precise approach for this problem. Even *love*, which does not presuppose anything, was separated out from these clusters, even though it has a very high frequency in these patterns.

To estimate the recall, we randomly sampled 200 verbs from our lexicon and asked two judges to label each verb to indicate if the verb presupposes an original eventuality. The judges agreed on 159 verbs of which twenty-seven were unanimously identified as presupposing an eventuality (13.5 per cent). We thus estimate that there are 217 verbs in our data that presuppose an eventuality (that is, 13.5 per cent of the 1,608 unique verbs in our data). As our system identified thirty-five verbs, we estimate the recall as 16 per cent. Thus, our proposed procedure is a high-precision, low-recall approach.

We see that some verbs appear in the Bad–Good class although they really belong in the Good–Good class. This was mostly due to some mislabeled words in the subjective lexicon and occasional misclassification of the verb *do*. However, the verbs with the highest probability in this class $\{forgive, pardon, excuse\}$ are correctly classified, and the other verbs are nonetheless good examples of the presupposition property.

Furthermore, most of the positive verbs discovered have the presupposition property *necessarily*, whereas many of the negative verbs only have this property *strongly* but not necessarily. We also see that there are fewer positive verbs with a

presupposition property, which suggests that people are generally altruistic (because a positive action does not necessarily presuppose that it was performed in return for something), and people are generally not negative unless prompted to be negative in return for another negative action (i.e., retaliation).

7 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. Specifically, we introduced an algorithm that semi-automatically discovers patterns encoding reciprocity based on a set of simple yet effective pronoun templates. We then ranked the identified patterns according to a scoring function and selected the most frequent ones. Using these patterns we queried the web and the Gutenberg corpus and extracted 13,443 reciprocal instances that represent a broad-coverage resource. Unsupervised clustering procedures were performed to generate meaningful semantic clusters of reciprocal instances. We also presented several extensions to these models (along with insightful observations) that incorporate meta-attributes like the verbs' affective value, study gender differences between participants, consider the textual context of the instances, and automatically discover verbs with certain presuppositions.

The experimental results provide nice insights into the problem, but can be further improved. For example, the pattern discovery procedure starts with the simplifying assumption that the participants to reciprocal eventualities are identified by personal pronouns. While this procedure ensures a high accuracy of the obtained patterns, it has a limited coverage. However, our pronoun templates were used just as a starting point to facilitate the discovery of the reciprocal patterns. Once these patterns are applied on text, they can capture reciprocal relationships between people as identified by any other named entities, provided we have a good tool that identifies the subject and the direct/indirect objects and a good named entity recognizer to identify people.

We also noticed that discovering polarity words is not always enough to capture the affect associated with 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 a future work, we will recluster the reciprocity pairs taking into consideration such speech acts.

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 and thanks for help. 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 that 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').

All these issues have to be studied in more detail. This kind of study is very important in the analysis of people's behavior, judgments, and thus their social interactions. One possibility, for example, would be to apply the reciprocity model for an in-depth empirical study of human sociocultural interactions in various text repositories in English or other natural languages. Thus, this research will provide a novel way of analyzing sociocultural interactions in language with direct application to the analysis of social groups and communities and it will bring new insights into the sociocultural aspects of these communities. The fields of sociology and behavior psychology have studied such issues for a long time. However, the connection to NLP has not been established yet.

Reciprocity is very important in studying other characteristics of social interaction as well. Recent discussions of the evolution of social intelligence, and of language itself, also place reciprocity at the center stage (Calvin and Bickerton 2000; Waal 2001). However, if theories of cognitive evolution are to draw on assumed reasoning about reciprocity, it is important to know which linguistic possibilities underlie this concept without restricting these models to simplistic notions drawn simply from a couple of constructions as done so far in the linguistics literature. The wide variety of ways English, and any other natural language for that matter, expresses reciprocity provides a rich resource for exploring alternative conceptualizations of this notion. Goody (1995), for example, claims that some of the ways languages encode reciprocity are motivated by speakers' models of the intentions of others, and the social relations they contract, in addition to their observed actions. Thus, we believe that any extension of the reciprocal model proposed in this paper should take such cognitive factors into account.

Furthermore, according to our preliminary experiments on English reciprocity, chained and asymmetrical extensions of reciprocal constructions are particularly frequent in representing certain types of co-operative social involvement, yet these factors have generally been neglected in the literature. Thus, a potential line of research would be to identify and analyze the sets of expressions that allow the formation of transitive chains of reciprocal behaviors and actions, which can be extracted from large text collections.

A further issue that has not been properly explored is the role of sociocultural models in licensing extensions of reciprocal relations in particular contexts. Semantic extensions like these build on culture-specific assumptions about the types of reciprocity, identity, and collectivity assertable of different social categories. Such a research direction would extend our knowledge of such constructions, and enable their investigation in further detail with wide empirical coverage.

Last, but not the least, we believe that this line of research has the potential to investigate the application of current NLP technologies and the development of new technologies for social sciences. Such a direction might have a transformative impact on social science research by enabling social scientists to use sophisticated NLP tools to analyze large data collections and test hypotheses, which would be difficult to test otherwise.

Acknowledgments

We would like to thank Chen Li for helping with the pattern ranking procedure. We are also grateful to the anonymous reviewers of this special issue of the *Natural Language Engineering* journal for their insightful comments. Finally, this work was partially inspired by Ray Jackendoff's invited talk on 'The Peculiar Logic of Value' given in September 2008 at the University of Illinois. The present paper is a much extended and revised version of Paul, Girju, and Li (2009).

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* **9**: 1981–2014.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. 2003. *An Introduction to MCMC for Machine Learning*.

Asher, N., and Lascarides, A. 2003. *Logics of Conversation*. Cambridge, England, UK: Cambridge University Press.

Baker, C., Fillmore, Ch., and Lowe, J. 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)*, Montreal, Quebec, Canada, pp. 86–90. Morristown, NJ: Association for Computational Linguistics.

Barwise, J., and Perry, J. 1985. Semantic innocence and uncompromising situations. In A. P. Martinich (ed.), *The Philosophy of Language*, pp. 401–413. New York: Oxford University Press.

Becker, L. (ed.) 1990. *Reciprocity*. Chicago, IL: University of Chicago Press.

Blei, D., Ng, A., and Jordan, M. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* **3**: 993–1022.

Calvin, W., and Bickerton, D. 2000. *Lingua ex Machina*. Cambridge, MA: MIT Press.

Chambers, N., and Jurafsky, D. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Empirical Methods in Natural Language Processi ng Conference (EMNLP)*, pp. 698–706.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP)*.

Chambers, N., Wang, S., and Jurafsky, D. 2007. Classifying temporal relations between events. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chang, J., and Blei, D. 2009. Relational topic models for document networks. In *AISTATS '09: Twelfth International Conference on Artificial Intelligence and Statistics*.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. 2009. Reading tea leaves: how humans interpret topic models. In *Neural Information Processing Systems*.

Chklovski, T., and Pantel, P. 2004. Verbocean: mining the web for fine-grained semantic verb relations. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) Conference*.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**(1): 37–46.

Connor, R. J., and Mosimann, J. E. 1969, March. Concepts of independence for proportions with a generalization of the dirichlet distribution. *Journal of the American Statistical Association* **64**(325): 194–206.

Dalrymple, M., Kazanawa, M., Kim, Y., Mchombo, S., and Peters, S. 1998. Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy* **21**: 159–210.

Davidov, D., and Rappoport, A. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automaticaly generated sat analogy questions. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Dempster, A. P., Laird, N. M., and Rdin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* **39**: 1–38.

De Waal, F. 2001. *Tree of Origin: What Primate Behavior can Tell Us About Human Social Evolution*. Cambridge, MA: Harvard University Press.

Dhillon, I. S., Guan, Y., and Kulis, B. 2007. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(11): 1944–1957.

Esuli, A., and Sebastiani, F. 2007. Pageranking wordnet synsets: an application to opinion mining. In *Proceedings of ACL-07, the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, pp. 424–431.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. 2004. Methods for domain-independent information extraction from the web: an experimental comparison. In *Proceedings of the National Conference on Artificial Intelligence (AAAI) Conference*.

Fehr, E., and Gachter, S. 2000. Cooperation and punishment in public goods experiments. *American Economic Review* **90**: 980–994.

Fellbaum, C. 1998. *WordNet – An Electronic Lexical Database*. Cambridge MA: MIT Press.

Gergen, K., Greenberg, M., and Willis, R. (eds.) 1980. *Social Exchange: Advances in Theory and Research*. New York: Plenum.

Gilks, W. R., Richardson, S., and D. J. Spiegelhalter. 1995. *Markov Chain Monte Carlo in Practice*. CRC Press.

Girju, R. 2010. Towards social causality: an analysis of interpersonal relations in online blogs and forums. In *Proceedings of ICWSM 2010 – International AAAI Conference on Weblogs and Social Media*. Association for the Advancement of Artificial Intelligence (AAAI).

Glickman, O., and Dagan, I. 2003. Identifying lexical paraphrases from a single corpus: a case study for verbs. In *International Conference Recent Advances of Natural Language Processing (RANLP)*.

Goody, E. 1995. *Social Intelligence and Interaction*. Cambridge, England, UK: Cambridge University Press.

Griffiths, T., and Steyvers, M. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*.

Grosz, B., and Sidner, C. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* **12**(3): 175–204.

Halpin, H., and Moore, J. D. 2006. Event extraction in a plot advice agent. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 857–864.

Haspelmath, M. 2007. Further remarks on reciprocal constructions. In P. Vladimir Nedjalkov (ed.), *Reciprocal Constructions*, pp. 2087–2115. Amsterdam, Netherlands: John Benjamins

Hearst, M. 1998. Automated Discovery of WordNet Relations. In C. Fellbaum (ed.), *An Electronic Lexical Database and Some of its Applications*, pp. 131–151. Cambridge, MA: MIT Press.

Heim, I. 1991. Reciprocity and plurality. *Linguistic Inquiry* **22**: 63–101.

Heinrich, G. 2008. Parameter estimation for text analysis. Technical Report, University of Leipzig.

Hobbs, J. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics* **22**(2): 181–209.

Hobbs, J., Stickel, M., Appelt, D., and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* **63**(1–2): 69–142.

Hofman, J., and Wiggins, C. 2008. Bayesian approach to network modularity. *Physical Review Letters* **100**(25): 258701.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA. ACM, pp. 50–57.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. 2006. OntoNotes: the 90% solution. In *NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York, NY, USA, pp. 57–60. Morristown, NJ: Association for Computational Linguistics.

Hughes, T., and Ramage, D. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June. Association for Computational Linguistics, pp. 581–589.

Jackendoff, R. 2005. The peculiar logic of value. *Journal of Cognition and Culture* **6**: 375–407.

Joanis, E., Stevenson, S., and James, D. 2008. A general feature space for automatic verb classification. *Natural Language Engineering* **14**(3): 337–367.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. 1998. Introduction to variational methods for graphical methods. In *Machine Learning*, pp. 183–233. Cambridge, MA: MIT Press.

Kemp, C., Tenenbaum, J., Griffiths, T., Yamada, T., and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

Kingsbury, P., Palmer, M., and Marcus, M. 2002. Adding semantic annotation to the Penn Treebank. In *Proceedings of the 2nd Human Language Technology Conference (HLT 2002)*, San Diego, California, pp. 252–256.

Kipper, K., H. Trang Dang, and Palmer, M. 2000. Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX, pp. 691–696.

König, E. 2005. Reciprocity in language: cultural concepts and patterns of encoding. *Uhlenbeck Lecture* 23, Amsterdam, The Netherlands. Amsterdam, Netherlands: Institute for Advanced Study.

Lehnert, W., Dyer, M., Johnson, P., Yang, C., and Harley, S. 1983. BORIS – an experiment in in-depth understanding of narratives. *Artificial Intelligence* **20**(1): 15–62.

Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago, IL: University of Chicago Press.

Li, W., Blei, D., and Mccallum, A. 2007. Nonparametric bayes pachinko allocation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Li, X., and Roth, D. 2001. Exploring evidence for shallow parsing. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pp. 107–110.

Lin, D., and Pantel, P. 2001. Discovery of inference rules for question answering. *Natural Language Engineering* **7**: 343–360.

Mandler, J. 1984. *Stories, Scripts and Scenes: Aspects of Schema Theory*. Hillsdale, NJ: Lawrence Erlbaum.

Maslova, E., and Nedjalkov, V. 2005. Reciprocal constructions. In M. Haspelmath, M. Dryer, D. Gill, and B. Comrie (eds.), *The World Atlas of Language Structures*, pp. 430–433. New York: Oxford University Press.

Mei, Q., Cai, D., Zhang, D., and Zhai, C. X. 2008. Topic modeling with network regularization. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, New York, NY, USA. ACM, pp. 101–110.

Merlo, P., and Stevenson, S. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics* **27**: 373–408.

Minnen, G., Carroll, J., and Pearce, D. 2000. Robust, applied morphological generation. In *INLG '00: Proceedings of the First International Conference on Natural Language Generation*, Morristown, NJ, USA. Association for Computational Linguistics, pp. 201–208.

Mitzenmacher, M., and Upfal, E. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. New York, NY: Cambridge University Press.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* **39**: 103–134.

Parkkinen, J., Gyenge, A., Sinkkonen, J., and Kaski, S. 2009. A block model suitable for sparse graphs. In: H. Blockeel, K. Borgwardt and X. Yan (eds.), *Proceedings of the 7th International Workshop on Mining and Learning with Graphs*, pp. 2–4. Belgium: Leuven.

Paul, M., and Girju, R. 2009. Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, Singapore. Association for Computational Linguistics.

Paul, M., Girju, R., and Li, C. 2009. Mining the web for reciprocal relationships. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp. 75–83.

Pustejovsky, J., and Verhagen, M. 2009. SemEval-2010 task 13: evaluating events, time expressions, and temporal relations (TempEval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, Boulder, Colorado. Association for Computational Linguistics, pp. 112–116.

Rabiner, L., and Juang, B. 1986. An introduction to hidden Markov models. *ASSP Magazine, IEEE* [see also *IEEE Signal Processing Magazine*] **3**(1): 4–16.

Ramage, D., Rosen, E., Chuang, J., Manning, C. D., and McFarland, D. A. 2009. Topic modeling for the social sciences. In *NIPS 2009 Workshop on Applications for Topic Models.*

Resnik, P. 1993. *Selection and Information: A Class-based Approach to Lexical Relationships.* Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.

Resnik, P., and Diab, M. 2000. Measuring verb similarity. In *12th Second Annual Meeting of the Cognitive Science Society (COGS CI).*

Sahlins, M., ed. 1972. *Stone Age Economics.* Chicago, IL: Aldine-Atherton.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.* Hillsdale, NJ: Lawrence Erlbaum.

Tsuruoka, Y., and Tsujii, J. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 467–474.

Turney, P. 2006. Similarity of semantic relations. *Computational Linguistics* **32**(3): 379–416.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. 2007. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic. Association for Computational Linguistics, pp. 75–80.

Wallach, H. M. 2006. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 977–984.

Webber, B., Knott, A., Stone, M., and Joshi, A. 2003. Anaphora and discourse structure. *Computational Linguistics* **29**(4): 545–588.

Wilson, T., Wiebe, J., and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology (HLT/EMNLP) Conference.*

Zanzotto, F. M., Pennacchiotti, M., and Pazienza, M. T. 2006. Discovering asymmetric entailment relations between verbs using selectional preferences. In *International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL).*

Zhai, C., Velivelli, A., and Yu, B. 2004. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD 22204*, pp. 743–748.