Automatic Detection and Classification of Social Events

Apoorv Agarwal
Department of Computer Science
Columbia University
New York, U.S.A.
apoorv@cs.columbia.edu

Owen Rambow
CCLS
Columbia University
New York, U.S.A.
rambow@ccls.columbia.edu

Abstract

In this paper we introduce the new task of *social event* extraction from text. We distinguish two broad types of social events depending on whether only one or both parties are aware of the social contact. We annotate part of Automatic Content Extraction (ACE) data, and perform experiments using Support Vector Machines with Kernel methods. We use a combination of structures derived from phrase structure trees and dependency trees. A characteristic of our events (which distinguishes them from ACE events) is that the participating entities can be spread far across the parse trees. We use syntactic and semantic insights to devise a new structure derived from dependency trees and show that this plays a role in achieving the best performing system for both social event detection and classification tasks. We also use three data sampling approaches to solve the problem of data skewness. Sampling methods improve the F1-measure for the task of relation detection by over 20% absolute over the baseline.

1 Introduction

This paper introduces a novel natural language processing (NLP) task, *social event extraction*. We are interested in this task because it contributes to our overall research goal, which is to extract a social network from written text. The extracted social network can be used for various applications such as summarization, question-answering, or the detection of main characters in a story. For example, we manually extracted the social network of characters in *Alice in Wonderland* and ran standard social network analysis algorithms on the network. The most influential characters in the story were correctly detected. Moreover, characters occurring in a scene together were given same social roles and positions. Social network extraction has recently been applied to literary theory (Elson et al., 2010) and has the potential to help organize novels that are becoming machine readable.

We take a "social network" to be a network consisting of individual human beings and groups of human beings who are connected to each other by the virtue of participating in social events. We define social events to be events that occur between people where at least one person is aware of the other and of the event taking place. For example, in the sentence *John talks to Mary*, entities John and Mary are aware of each other and the talking event. In the sentence *John thinks Mary is great*, only John is aware of Mary and the event is the thinking event. In the sentence *Rabbit ran by Alice* there is no evidence about the cognitive states of Rabbit and Alice (because the Rabbit could have run by Alice without any one of them noticing each other). A text can describe a social network in two ways: explicitly, by stating the type of relationship between two individuals (e.g. husband-wife), or implicitly, by describing an event which creates or perpetuates a social relationship (e.g. *John talked to Mary*). We will call these types of events *social events*. We define two types of social events: **interaction**, in which both parties are aware of the social event (e.g., a conversation), and **observation**, in which only one party is aware of the interaction (e.g., thinking about or

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1024–1034,
MIT, Massachusetts, USA, 9-11 October 2010.

spying on someone). Note that the notion of cognitive state is crucial to our definition. This paper is the first attempt to detect and classify social events present in text.

Our task is different from related tasks, notably from the Automated Content Extraction (ACE) relation and event extraction tasks because the events are different (they are a class of events defined through the effect on participants' cognitive state), and the linguistic realization is different. Mentions of entities[1] engaged in a social event are often quite distant from each other in the sentence (unlike in ACE relations where about 70% of relations are local, in our social event annotation, only 25% of the events are local. In fact, the average number of words between entities participating in any social event is 9.)

We use tree kernel methods (on structures derived from phrase structure trees and dependency trees) in conjunction with Support Vector Machines (SVMs) to solve our tasks. For the design of structures and type of kernel, we take motivation from a system proposed by Nguyen et al. (2009) which is a state-of-the-art system for relation extraction. Data skewness turns out to be a big challenge for the task of relation detection since there are many more pairs of entities without a relation as compared to pairs of entities that have a relation. In this paper we discuss three data sampling techniques that deal with this skewness and allow us to gain over 20% in F1-measure over our baseline system. Moreover, we introduce a new sequence kernel that outperforms previously proposed sequence kernels for the task of social event detection and plays a role to achieve the best performing system for the task of social event detection and classification.

The paper is structured as follows. In Section 2, we compare our work to existing work, notably the ACE extraction literature. In Section 3, we present our task in detail, and explain how we annotated our corpus. We also show why this is a novel task, and how it is different from the ACE extraction tasks. We then discuss kernel methods and the structures we use, and introduce our new structure in Section 4. In Section 5, we present the sampling methods used for experiments. In Section 6 we present our experiments and results for social event detection and social event classification tasks. We conclude in Section 7 and mention our future direction of research.

[1] An *entity mention* is a reference of an entity in text. Also, we use *entities* and *people* interchangeably since the only entities we are interested in are people or groups of people.

2 Literature Survey

There has not been much work in developing techniques for ACE event extraction as compared to ACE relation extraction. The most salient work for event extraction is Grishman et al. (2005) and Ji and Grishman (2008). To solve the task for event extraction, Grishman et al. (2005) mainly use a combination of pattern matching and statistical modeling techniques. They extract two kinds of patterns: 1) the sequence of constituent heads separating anchor and its arguments and 2) a predicate argument subgraph of the sentence connecting anchor to all the event arguments. In conjunction they use a set of Maximum Entropy based classifiers for 1) Trigger labeling, 2) Argument classification and 3) Event classification. Ji and Grishman (2008) further exploit a correlation between senses of verbs (that are triggers for events) and topics of documents.

Our work shares some similarities. However, instead of building different classifiers, we use kernel methods with SVMs that "naturally" combine various patterns. The structures we use for kernel methods are a super-set of the patterns used by Grishman et al. (2005). Moreover, in our work, we take gold annotation for entity mentions, and do not deal with the task of named entity detection or resolution. Finally, our social events are a broad class of event types, and they involve linguistic expressions for expressing interactions and cognition that do not seem to have a correlation with the topics of documents.

There has been much work in extracting ACE relations. The supervised approaches used for relation extraction can broadly be divided into three main categories: 1) feature-based approaches 2) kernel-based approaches and 3) a combination of feature and kernel based approaches. The state-of-the-art feature based approach is that of GuoDong et al. (2005). They use diverse lexical, syntactic and semantic knowledge for the task. The lexical features they use are words between, before, and after target entity mentions, the type of entity (Person, Organization etc.), the type of mention (named, nominal or pronominal) and a feature called overlap

that counts the number of other entity mentions and words between the target entities. To incorporate syntactic features they use features extracted from base phrase chunking, dependency trees and phrase structure trees. To incorporate semantic features, their approach uses resources like a country list and WordNet. GuoDong et al. (2005) report that 70% of the entities are embedded within each other or separated by just one word. This is a major difference to our task because most of our relations span over a long distance in a sentence.

Collins and Duffy (2002) are among the earliest researchers to propose the use of tree kernels for various NLP tasks. Since then kernels have been used for the task of relation extraction (Zelenko et al., 2002; Zhao and Grishman, 2005; Zhang et al., 2006; Moschitti, 2006b; Nguyen et al., 2009). For an excellent review of these techniques, see Nguyen et al. (2009). In addition, there has been some work that combines feature and kernel based methods (Harabagiu et al., 2005; Culotta and Jeffrey, 2004; Zhou et al., 2007). Apart from using kernels over dependency trees, Culotta and Jeffrey (2004) incorporate features like words, part of speech (POS) tags, syntactic chunk tag, entity type, entity level, relation argument and WordNet hypernym. Harabagiu et al. (2005) leverage this approach by adding more semantic feature derived from semantic parsers for FrameNet and PropBank. Zhou et al. (2007) use a context sensitive kernel in conjunction with features they used in their earlier publication (GuoDong et al., 2005). However, we take an approach similar to Nguyen et al. (2009). This is because it incorporates many of the features suggested in feature-based approaches by using combinations of various structures derived from phrase structure trees and dependency trees. In addition we use data sampling techniques to deal with the problem of data skewness. We not only try the structures suggested by Nguyen et al. (2009) but also introduce a new sequence structure on dependency trees. We discuss their structures and kernel method in detail in Section 4.

3 Social Event Annotation Data

3.1 Social Event Annotation

There has been much work in the past on annotating entities, relations and events in free text, most notably the ACE effort (Doddington et al., 2004). We leverage this work by annotating social events on the English part of ACE 2005 Multilingual Training Data[2] that has already been annotated for entities, relations and events. In Agarwal et al. (2010), we introduce a comprehensive set of social events which are conceptually different from the event annotation that already exists for ACE. Since our annotation task is complex and layered, in Agarwal et al. (2010) we present confusion matrices, Cohen's Kappa, and F-measure values for each of the decision points that the annotators go through in the process of selecting a type and subtype for an event. Our annotation scheme is reliable, achieving a moderate kappa for relation detection (0.68) and a high kappa for relation classification (0.86). We also achieve a high global agreement of 69.7% using a measure which is inspired by Automated Content Extraction (ACE) inter-annotator agreement measure. This compares favorably to the ACE annotation effort.

Following are the two broad types of social events that were annotated:

Interaction event (INR): When both entities participating in an event are aware of each other and of the social event, we say they have an INR relation. Consider the following Example (1).

(1) [Toujan Faisal], 54, {said} [she] was {informed} of the refusal by an [Interior Ministry committee] overseeing election preparations. INR

As is intuitive, if one person *informs* the other about something, both have to be cognizant of each other and of the *informing* event in which they are both participating.

Observation event (OBS): When only one person (out of the two people that are participating in an event) is aware of the other and of the social event, we say they have an OBS relation. Of the type OBS, there are three subtypes: Physical Proximity (PPR), Perception (PCR) and Cognition (COG). PPR requires that one entity can observe the other entity in real time not through a broadcast medium, in contrast to the subtype PCR, where one entity observes the other through media (TV, radio, magazines etc.) Any other observation event that is not PPR or PCR

[2] Version: 6.0, Catalog number: LDC2005E18

is COG. Consider the aforementioned Example (1). In this sentence, the event *said* marks a COG relation between **Toujan Faisal** and the **committee**. This is because, when one person talks about another person, the other person must be present in the first person's cognitive state without any requirement on physical proximity or external medium.

As the annotations revealed, PPR and PCR occurred only twice and once, respectively, in the part of ACE corpus we annotated. (They occur more frequently in another genre we are investigating such as literary texts.) We omit these extremely low-frequency categories from our current study; in this paper we build classifiers to detect and classify only INR and COG events.

3.2 Comparison Between Social Events and ACE Annotations

The ACE effort is about entity, relation and event annotation. We use their annotations for entity types PER.Individual and PER.Group and add our social event annotations. Our event annotations are different from ACE event annotations because we annotate text that expresses the cognitive states of the people involved, or allows the annotator to infer it. Therefore, at the top level of classification we differentiate between events in which only one entity is cognizant of the other (observation) versus events when both entities are cognizant of each other (interaction). This distinction is, we believe, novel in event or relation annotation.

Now we present statistics and examples to make clear how our annotations are different from ACE event annotations. The statistics are based on 62 documents from the ACE corpus. These files contain a total of 212 social events. We found a total of 63 candidate ACE events that had at least two Person entities involved. Out of these 63 candidate events, 54 match our annotations. The majority of social events that match the ACE events are of type INR. On analysis, we found that most of these correspond to the ACE event type CONTACT. Specifically, the "meeting" event, which is an ACE CONTACT event and an INR event according to our definition, is the major cause of overlap. However, our type INR has a broader definition than ACE type CONTACT. For example, in Example 1, we recorded an INR event between **Toujan Faisal** and **committee** (event span: *informed*). ACE does not record any event between these two entities because *informed* does not entail a CONTACT event for ACE event annotations. Another example that will clarify the difference is the following:

(2) In central Baghdad, [a Reuters cameraman] and [a cameraman for Spain's Telecinco] died when an American tank fired on the Palestine Hotel

ACE has annotated the above example as an event of type CONFLICT in which there are two entities that are of type person: the **Reuters cameraman** and the **cameraman for Spain's Telecinco**, both of which are arguments of type "Victim". Being an event that has two person entities involved makes the above sentence a potential social event. However, we do not record any event between these entities since the text does not reveal the cognitive states of the two entities; we do not know whether one was aware of the other.

ACE defines a class of social relations (PER-SOC) that records named relations like friendship, co-worker, long lasting etc. Also, there already exist systems that detect and classify these relations well. Therefore, even though these relations are directly relevant to our overall goal of social event extraction, we do not annotate, detect or classify these relations in this paper.

4 Tree Kernels, Discrete Structures, and Language

In this section, we give details of the structures and kernel we use for our classification tasks. We also discuss our motivation behind using these methods. Linear learning machines are one of the most popular machines used for classification problems. The objective of a typical classification problem is to learn a function that separates the data into different classes. The data is usually in the form of features extracted from abstract objects like strings, trees, etc. A drawback of learning by using complex functions is that complex functions do not generalize well and thus tend to over-fit. The research community therefore prefers linear classifiers over other complex classifiers. But more often than not, the data is not linearly separable. It can be made linearly separable by increasing the dimensionality of data but then learning suffers from the curse of

dimensionality and classification becomes computationally intractable. This is where kernels come to the rescue. The well-known kernel trick aids us in finding similarity between feature vectors in a high dimensional space without having to write down the expanded feature space. The essence of kernel methods is that they compare two feature vectors in high dimensional space by using a dot product that is a function of the dot product of feature vectors in the lower dimensional space. Moreover, Convolution Kernels (first introduced by Haussler (1999)) can be used to compare abstract objects instead of feature vectors. This is because these kernels involve a recursive calculation over the "parts" of a discrete structure. This calculation is usually made computationally efficient using Dynamic Programming techniques. Therefore, Convolution Kernels alleviate the need of feature extraction (which usually requires domain knowledge, results in extraction of incomplete information and introduces noise in the data). Therefore, we use convolution kernels with a linear learning machine (Support Vector Machines) for our classification task.

Now we present the "discrete" structures followed by the kernel we used. We use the structures previously used by Nguyen et al. (2009), and propose one new structure. Although we experimented with all of their structures,[3] here we only present the ones that perform best for our classification task. All the structures and their combinations are derived from a variation of the underlying structures, Phrase Structure Trees (PST) and Dependency Trees (DT). For all trees we first extract their Path Enclosed Tree, which is the smallest common subtree that contains the two target entities (Moschitti, 2004). We use the Stanford parser (Klein and Manning, 2003) to get the basic PSTs and DTs. Following are the structures that we refer to in our experiments and results section:

PET: This refers to the smallest common phrase structure tree that contains the two target entities.

Dependency Words (DW) tree: This is the smallest common dependency tree that contains the two target entities. In Figure 1, since the target entities are at the leftmost and rightmost branch of the dependency tree, this is in fact a DW (ignoring the grammatical relations on the arcs).

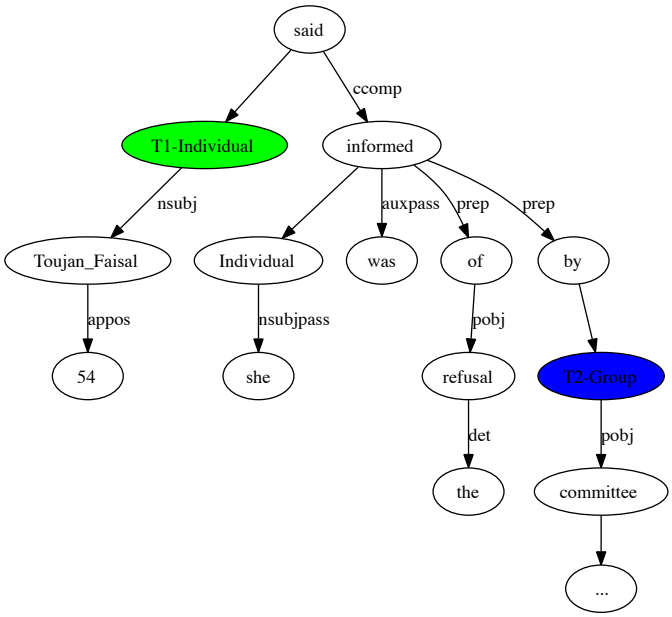

Figure 1: Dependency parse tree for the sentence (in the ACE corpus): "[Toujan Faisal], 54, {said} [she] was {informed} of the refusal by an [Interior Ministry committee] overseeing election preparations."

Grammatical Relation (GR) tree: If we replace the words at the nodes by their relation to their corresponding parent in DW, we get a GR tree. For example, in Figure 1, replacing *Toujan_Faisal* by *nsubj*, *54* by *appos*, *she* by *nsubjpass* and so on.

Grammatical Relation Word (GRW) tree: We get this tree by adding the grammatical relations as separate nodes between a node and its parent. For example, in Figure 1, adding *nsubj* as a node between *T1-Individual* and *Toujan_Faisal*, *appos* as a node between *54* and *Toujan_Faisal*, and so on.

Sequence Kernel of words (SK1): This is the sequence of words between the two entities, including their tags. For our example in Figure 1, it would be *T1-Individual Toujan Faisal 54 said she was informed of the refusal by an T2-Group Interior Ministry committee*.

Sequence in GRW tree (SqGRW): This is the new structure that we introduce which, to the best of

[3] We omitted SK6, which is the worst performing sequence kernel in (Nguyen et al., 2009).

our knowledge, has not been used before for similar tasks. It is the sequence of nodes from one target to the other in the GRW tree. For example, in Figure 1, this would be *Toujan_Faisal nsubj T1-Individual said ccomp informed prep by T2-Group pobj committee*.

We also use combinations of these structures (which we refer to as "combined-structures"). For example, PET_GR_SqGRW means we used the three structures (PET, GR and SqGRW) together with a kernel that calculates similarity between forests.

We use the Partial Tree (PT) kernel, first proposed by Moschitti (2006a), for structures derived from dependency trees and Subset Tree (SST) kernel, proposed by Collins and Duffy (2002), for structures derived from phrase structure trees. PT is a relaxed version of the SST; SST measures the similarity between two PSTs by counting all subtrees common to the two PSTs. However, there is one constraint: all daughter nodes of a node must be included. In PTs this constraint is removed. Therefore, in contrast to SSTs, PT kernels compare many more substructures. They have been used successfully by (Moschitti, 2004) for the task of semantic role labeling.

The choices we have made are motivated by the following considerations. We are interested in modeling classes of events which are characterized by the cognitive states of participants–who is aware of whom. The predicate-argument structure of verbs can encode much of this information very efficiently, and classes of verbs express their predicate-argument structure in similar ways. For example, many verbs of communication can express their arguments using the same pattern: *John talked/spoke/lectured/ranted/testified to Mary about Percy*. Independently of the verb, **John** is in a COG relation with **Percy** and in an INR relation with **Mary**. All these verbs allow us to drop either or both of the prepositional phrases, without altering the interpretation of the remaining constituents. And even more strikingly, any verb that can be put in that position is likely to have this interpretation; for example, we are likely to interpret the neologistic *John gazooked to Mary about Percy* as a similarly structured social event.

The regular relation between verb alternations and meaning components has been extensively studied (Levin, 1993; Schuler, 2005). This regularity in the syntactic predicate-argument structure allows us to overcome lexical sparseness. However, in order to exploit such regularities, we need to have access to a representation which makes the predicate-argument structure clear. Dependency representations do this. Phrase structure representations also represent predicate-argument structure, but in an indirect way through the structural configurations, and we expect this to increase the burden on the learner. (In some phrase structure representations, some arguments and adjuncts are not disambiguated.) When using dependency structures, the SST kernel is far less appealing, since it forces us to always consider all daughter nodes of a node. However, as we have seen, it is certain daughter nodes, such as the presence of a *to* PP and a *about* PP, which are important, while other daughters, such as temporal or locative adjuncts, should be disregarded. The PT kernel allows us to do this.

5 Sampling Methods

In this section we present the data sampling methods we use to deal with data skewness. We employ two well-known data sampling methods on the training data before creating a model for test data; random under-sampling and random over-sampling (Kotsiantis et al., 2006; Japkowicz, 2000; Weiss and Provost, 2001). These techniques are non-heuristic sampling methods that aim at balancing the class proportions by removing examples of the majority class and by duplicating instances of the minority class respectively. The reason for using these techniques is that learning is usually optimized to achieve high accuracy. Therefore, when presented with skewed training data, a classifier may learn the target concept with a high accuracy by only predicting the majority class. But if one looks at the precision, recall, and F-measure, of such a classifier, they will be very low for the minority class. Since, like other researchers, we are evaluating the goodness of a model based on its precision, recall and F-measure and not on the accuracy on the test set, either we should change the optimization function of the classifier or employ data sampling techniques. We employ the latter because by balancing the class ratio, we are presenting the classifier with a more challenging task of achieving a good accuracy when the

majority base class is about 50%. The major drawbacks of the two techniques is that under-sampling throws away important information whereas over-sampling is prone to over-fitting (due to data duplication). As our results show, throwing away information about the majority class is much better than the system that tries to learn in an unbalanced scenario, but it performs worse than an approach using data duplication. Since we are using SVMs as a classifier, over-fitting is unlikely as reported by Kolcz et al. (2003).

In order to be sure that we are not over-fitting, we tried another sampling method proposed by Ha and Bunke (1997), which is shown to be good solution to avoid over-fitting by Chawla et al. (2002). This sampling technique proposes to generate synthetic examples of the minority class by "perturbing" the training data. Specifically, Ha and Bunke (1997) produced new synthetic examples for the task of handwritten character recognition by doing operations like rotation and skew on characters. The basic idea is to produce synthetic examples that are "close" to the real example from which these synthetic points are generated. Analogously, we tried two transformations on our dependency tree structures to produce synthetic examples. The first transformation is based on the observation that in control verb constructions, the matrix verb typically does not contribute to the interpretation as a social event or not. In this transformation, we lower the subject to an argument verb if it does not have a subject, and repeat this procedure iteratively. As it turned out, this transformation only occurred 15 times, and therefore it does not serve the purpose of over-sampling. We tried a more relaxed transformation on the rightmost target in the tree. Here, the observation is that for the COG social events, the second target may be very deeply embedded in the tree. For example, in Example 1, Toujan Faisal and the Interior Ministry Committee participate in a COG event (because Faisal is aware of the Committee during the saying event). However, the contents of what Faisal said is only relevant to the extent that it pertains to the committee. The depth of the embedding of the second target creates issues of data sparseness, as the path-enclosed trees become very large and very diverse. Our transformation, therefore, is to move the second target to its grandmother node, attaching it on the left, and to recalculate the path-enclosed tree, which is now smaller. This is repeated iteratively, so that a sentence with a deeply embedded second target can yield a large number of synthesized structures.

6 Experiments And Results

In this section we present experiments and results for our two tasks: social event detection and classification. For the social event detection task, we wish to validate the following research hypotheses. First, we aim to show the importance of using data sampling when evaluating on F-measure; specifically, we expect under-sampling to outperform no sampling, over-sampling to outperform under-sampling, and over-sampling with transformations to out perform over-sampling without transformations. In contrast, the social event classification task does not suffer from data skewness because the INR and COG relations; both occur almost the same number of times. Therefore, sampling methods may not be applied for this task. Second, for both tasks, we expect that a combination of kernels will out-perform individual kernels. Moreover, we expect that dependency trees will have a crucial role in achieving the best performance.

6.1 Experimental Set-up

We use part of ACE data that we annotated for social events. In all, we annotated 138 ACE documents. We retained the ACE entity annotations. We consider all entity mention pairs in a sentence. If our annotators recorded a relation between a pair of entity mentions, we say there is a relation between the corresponding entities. If there are any other pairs of entity mentions for the same pair of entity, we discard those. For all other pairs of entity mentions, we say there is no relation. Out of 138 files, four files did not have any positive or negative examples (because there were very few and sparse entity mentions in these four files). We found a total of 1291 negative examples, 172 examples belonging to class INR and 174 belonging to class COG.

We use Jet's sentence splitter[4] and the Stanford Parser (Klein and Manning, 2003) for phrase structure trees and dependency parses. For classifica-

[4] http://cs.nyu.edu/grishman/jet/jetDownload.html

tion, we used Alessandro Moschitti's SVM-Light-TK package (Moschitti, 2006b) which is built on the SVM-Light implementation of Joakhims (1999). For all our experiments, we perform 5-fold cross-validation. We randomly divide the whole corpus into 5 equal parts, such that no news story (or document) gets divided among two parts. For each fold, we then merge 4 parts to create a training corpus and treat the remaining part as a test corpus. By keeping individual news stories intact, we make sure that vocabulary specific to one story does not unrealistically improve the performance.

6.2 Social Event Detection

Social event detection is the task of detecting if any social event exists between a pair of entities in a sentence. We formulate the problem as a binary classification task by labeling an example that does not have a social event as class -1 and by labeling an example that either has an INR or COG social event as class 1. First we present results for our baseline system. Our baseline system uses various structures and their combinations but without any data balancing. [5]

Kernel	P	R	F1
PET	70.28	21.46	32.38
GR	**87.79**	15.21	25.55
GRW	76.42	8.26	14.8
SqGRW	48.78	6.08	10.38
PET_GR	70.21	**27.76**	**38.89**
PET_GR_SqGRW	71.06	26.74	38.02
GR_SqGRW	82.0	24.47	36.12
GRW_SqGRW	68.19	17.01	25.06
GR_GRW_SqGRW	79.81	21.99	32.57

Table 1: Baseline System for the task of social event detection. The proportion of positive data in training and test set is 21.1% and 20.6% respectively.

Table 1 presents results for our baseline system. Grammatical relation tree structure (GR), a structure derived from dependency tree by replacing the words by their grammatical relations achieves the best precision. This is probably because the classifier learns that if both the arguments of a predicate contain target entities then it is a social event. Among kernels for single structures, the path enclosed tree for PSTs (PET) achieves the best recall. Furthermore, a combination of structures derived from PSTs and DTs performs best. The sequence kernels, perform much worse than SqGRW (F1-measure as low as 0.45). Since it is the same case for all subsequent experiments, we omit them from the discussion.

[5] Although we experimented with many more structures and their combinations, due to space restrictions we mention only the top results.

Kernel	P	R	F1
PET	28.89	77.06	41.96
GR	**35.68**	72.47	47.37
GRW	29.7	83.6	43.6
SqGRW	34.31	**84.15**	**48.61**
PET_GR	34.38	83.94	**48.52**
PET_GR_SqGRW	34.34	83.66	**48.52**
GR_SqGRW	33.45	81.73	47.27
GRW_SqGRW	32.87	**84.44**	47.11
GR_GRW_SqGRW	32.73	83.26	46.82

Table 2: Under-sampled system for the task of relation detection. The proportion of positive examples in the training and test corpus is 50.0% and 20.6% respectively.

We now turn to experiments involving sampling. Table 2 presents results for under-sampling, i.e. randomly removing examples belonging to the negative class until its size matches the positive class. Table 2 shows a large gain in F1-measure of 9.72% absolute over the baseline system (Table 1). We found that worst performing kernel with under-sampling is SK1 with an F1-measure of 39.2% which is better than the best performance without under-sampling. These results make it clear that doing under-sampling greatly improves the performance of the classifier, despite the fact that we are using less training data (fewer negative examples). This is as expected because we are evaluating on F1-measure and the classifier is optimizing for accuracy.

Table 3 presents results for over-sampling i.e. replicating positive examples to achieve an equal number of examples belonging to the positive and negative class. Table 3 shows that the gain over the baseline system now is 22.2% absolute. Also, the gain over the under-sampled system is 12.5%

Kernel	P	R	F1
PET	50.9	57.21	53.62
GR	43.57	67.21	52.59
GRW	46.05	64.15	53.31
SqGRW	42.4	**72.75**	53.5
PET_GR	56.42	66.2	60.63
PET_GR_SqGRW	**57.28**	66.26	**61.11**
GR_SqGRW	44.35	71.17	54.52
GRW_SqGRW	44.77	68.79	54.12
GR_GRW_SqGRW	46.79	71.54	56.45

Table 3: Over-sampled system for the task of relation detection. The proportion of positive examples in the training and test corpus is 50.0% and 20.6% respectively.

absolute. As in the baseline system, a combination of structures performs best. As in the under-sampled system, when the data is balanced, SqGRW (sequence kernel on dependency tree in which grammatical relations are inserted as intermediate nodes) achieves the best recall. Here, the PET and GR kernel perform similar: this is different from the results of (Nguyen et al., 2009) where GR performed much worse than PET for ACE data. This exemplifies the difference in the nature of our event annotations from that of ACE relations. Since the average distance between target entities in the surface word order is higher for our events, the phrase structure trees are bigger. This means that implicit feature space is much sparser and thus not the best representation.

PET	37.04	66.49	47.28
GR	40.39	71.14	51.27
GRW	45.16	66.82	53.47
SqGRW	42.88	70.67	53.22
PET_GR	45.33	70.26	54.71
PET_GR_SqGRW	45.26	**72.97**	**55.67**
GR_SqGRW	43.73	71.47	54.06
GRW_SqGRW	45.70	71.30	55.32
GR_GRW_SqGRW	**45.91**	71.90	**55.70**

Table 4: Over-sampled System with transformation for relation detection. The proportion of positive examples in the training and test corpus is 51.7% and 20.6% respectively.

Table 4 presents results for using the over-sampling method with transformation that produces synthetic positive examples by using a transformation on dependency trees such that the new synthetic examples are "close" to the original examples. This method achieves a gain 16.78% over the baseline system. We expected this system to perform better than the over-sampled system but it does not. This suggests that our over-sampled system is not over-fitting; a concern with using oversampling techniques.

6.3 Social Event Classification

For the social event classification task, we only consider pairs of entities that have an event. Since these events could only be INR or COG, this is a binary classification problem. However, now we are interested in both outcomes of the classification, while earlier we were only interested in knowing how well we were finding relations (and not in how well we were finding "non-relations"). Therefore, accuracy is the relevant metric (Table 5).

Kernel	Acc
PET	76.85
GR	71.04
GRW	76.22
SqGRW	75.78
PET_GR	76.34
PET_GR_SqGRW	**78.72**
GR_SqGRW	75.60
GRW_SqGRW	76.96
GR_GRW_SqGRW	77.29

Table 5: System for the task of relation classification. The two classes are INR and COG, and we evaluate using accuracy (Acc.). The proportion of INR relations in training and test set is 49.7% and 49.63% respectively.

Even though the task of reasoning if an event is about one-way or mutual cognition seems hard, our system beats the chance baseline by 28.72%. These results show that there are significant clues in the lexical and syntactic structures that help in differentiating between interaction and cognition social events. Once again we notice that the combination of kernels works better than single kernels

alone, though the difference here is less pronounced. Among the combined-structure approaches, combinations with dependency-derived structures continue to outperform those not including dependency (the best all-phrase structure performer is PET_SK1 with 75.7% accuracy, not shown in Table 5).

7 Conclusion And Future Work

In this paper, we have introduced the novel tasks of social event detection and classification. We show that data sampling techniques play a crucial role for the task of relation detection. Through oversampling we achieve an increase in F1-measure of 22.2% absolute over a baseline system. Our experiments show that as a result of how language expresses the relevant information, dependency-based structures are best suited for encoding this information. Furthermore, because of the complexity of the task, a combination of phrase based structures and dependency-based structures perform the best. This revalidates the observation of Nguyen et al. (2009) that phrase structure representations and dependency representations add complimentary value to the learning task. We also introduced a new sequence structure (SqGRW) which plays a role in achieving the best accuracy for both, social event detection and social event classification tasks.

In the future, we will use other parsers (such as semantic parsers) and explore new types of linguistically motivated structures and transformations. We will also investigate the relation between classes of social events and their syntactic realization.

Acknowledgments

The work was funded by NSF grant IIS-0713548. We thank Dr. Alessandro Moschitti and Truc-Vien T. Nguyen for helping us with re-implementing their system. We acknowledge Boyi Xie for his assistance in implementing the system. We would also like to thank Dr. Claire Monteleoni and Daniel Bauer for useful discussions and feedback.

References

Apoorv Agarwal, Owen Rambow, and Rebecca J Passonneau. 2010. Annotation scheme for social network extraction from text. In *Fourth Linguistic Annotation Workshop, ACL*.

N V Chawla, L O Hall, K W Bowyer, and W P Kegelmeyer. 2002. Smote: Synthetic minority oversampling technique. In *Journal of Artificial Intelligence Research*.

M. Collins and N. Duffy. 2002. Convolution kernels for natural language. In *Advances in neural information processing systems*.

Aron Culotta and Sorensen Jeffrey. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 423–429, Barcelona, Spain, July.

G Doddington, A Mitchell, M Przybocki, L Ramshaw, S Strassel, and R Weischedel. 2004. The automatic content extraction (ace) program–tasks, data, and evaluation. *LREC*, pages 837–840.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden.

Ralph Grishman, David Westbrook, and Adam Meyers Proc. 2005. Nyu's english ace 2005 system description. In *ACE Evaluation Workshop*.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics*.

T. M. Ha and H Bunke. 1997. Off-line, handwritten numerical recognition by perturbation method. In *Pattern Analysis and Machine Intelligence*.

Sanda Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. 2005. Shallow semantics for relation extraction. In *International Joint Conference On Artificial Intelligence*.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz.

Nathalie Japkowicz. 2000. Learning from imbalanced data sets: Comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through unsupervised cross-document inference. In *Proceedings of ACL*.

Thorsten Joakhims. 1999. Making large-scale svm learning practical. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*.

Dan Klein and Chistopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS)*.

Aleksander Kolcz, Abdur Chowdhury, and Joshua Alspector. 2003. Data duplication: An imbalance problem. In *Workshop on Learning from Imbalanced Datasets, ICML*.

Sotiris Kotsiantis, Dimitris Kanellopoulos, and Panayiotis Pintelas. 2006. Handling imbalanced datasets: A review. In *GESTS International Transactions on Computer Science and Engineering*.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. The University of Chicago Press.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Conference on Association for Computational Linguistic*.

Alessandro Moschitti. 2006a. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*.

Alessandro Moschitti. 2006b. Making tree kernels practical for natural language learning. In *Proceedings of European chapter of Association for Computational Linguistics*.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. *Conference on Empirical Methods in Natural Language Processing*.

Karin Kipper Schuler. 2005. *Verbnet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, upenncis.

Gary M Weiss and Foster Provost. 2001. The effect of class distribution on classifier learning: an empirical study. Technical Report ML.TR-44, Rutgers University, August.

D. Zelenko, C. Aone, and A. Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of the EMNLP*.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of COLING-ACL*.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Meeting of the ACL*.

GuoDong Zhou, Min Zhang, DongHong Ji, and QiaoMing Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of EMNLP-CoNLL*.