Kernel-based Relation Extraction from Investigative Data

Cristina Giannone
CM Sistemi s.p.a and
University of Roma, Tor Vergata
Rome, Italy
cristina.giannone@gruppocm.it

Roberto Basili
University of Roma, Tor Vergata
Roma, Italy
basili@info.uniroma2.it

Chiara Del Vescovo
CM Sistemi s.p.a.
Roma, Italy
chiara.delvescovo@gruppocm.it

Paolo Naggar
CM Sistemi s.p.a.
Roma, Italy
paolo.naggar@gruppocm.it

Alessandro Moschitti
University of Trento
Trento, Italy
moschitti@disi.unitn.it

ABSTRACT

In a specific process of business intelligence, i.e. investigation on organized crime, empirical language processing technologies can play a crucial role. In the data used on investigative activities, such as police interrogatory or electronic eavesdropping and wiretap, it is customary to find out expressions in non conventional languages as dialects, slangs or coded words. The recognition and storage of complex relations among subjects mentioned in these sources is a very difficult and time consuming task, ultimately based on pools of experts. We discuss here an inductive relation extraction platform that opens the way to much cheaper and consistent workflows. SVMs here are employed to produce a set of possible interpretations for domain relevant concepts. An ontology population process is here realized, where further reasoning can be applied to proof the overall consistency of the extracted information. The empirical investigation presented here shows that accurate results, comparable to the expert teams, can be achieved, and parametrization allows to fine tune the system behavior for fitting the specific domain requirements.

1. ANALYSIS OF INVESTIGATIVE TEXTS

The semi automated extraction of information from data of a textual nature has become of interest in recent years for different theoretical and applicative contexts, relevant both in the investigative and decisional stages of judicial processes. Starting from the results of the research project "*ASTREA*, Information and Communication for Justice" [1], the *REVEAL* (Relation Extraction for inVEstigating criminAL enterprises) project has been carried out for setting up a relation extraction system and putting it on trial with unstructured judicial documents (such as questioning/confession reports or wiretap transcriptions). The own proper nature of documents generates noise due to the physical communication channels as the language quoted in the documents. This kind of texts, in fact, typically comprises spelling errors, ad-hoc abbreviations as well as incorrect punctuation and malformed sentences. Hence text mining techniques based on pure linguistic strategies fail to extract information from texts. The huge amount of texts involved in the investigation process needs the automation of the recognition of the specific relations. Currently the system population process is executed by teams of analysts that, after reading a document, annotate all quotations about facts and involved subjects on the base of a conceptual schema. This kind of analysis process produces an unavoidable bottleneck in the investigation process, the amount of daily texts produced by Italian Public Prosecutor's Offices is too large to be suitably managed through manual procedures in a short time. This affects the scalability and timeliness of the resulting operational procedures. Machine learning methods are highly beneficial in this respect. First, statistical learning methods represent the state-of-the-art in several Information Extraction and Semantic Parsing tasks as systematic benchmarking in the NLP area shown in international challenges ([9]), also providing robustness techniques to extract information from noisy text. Second, the adoption of inductive methods enables an incremental approach where the interleaving between the automatic learning for tagging relations and human validation allows us to scale up in a much cheaper fashion: at the i-th iteration, professional analysts may more quickly provide novel examples through the corrections of mistakes made by the system, and this triggers a novel stage $(i + 1)$ with a larger training datasets. The project aims to substitute the manual *analysis phase* employing REVEAL system for real-time extraction of information.

In this paper the first experiences in using the system and a comprehensive set of validating results are reported. This work only addresses the relation extraction process, since in the first phase of the document analysis we assume that quotations regarding entities are already made available. In particular, the reported experiments focus on texts whose target entity quotations have been already manually tagged by domain experts. This work presents the main results of the *REVEAL* project and, in particular, how the prototype automatizes the relation extraction stage. As we will see, the proposed methodology has been successfully applied in the project and early results confirm the wide applicability of the proposed approach also in other noisy data domains.

1.1 Definition of the task

The set of documents relevant for the typical analysis conducted in this domain are characterized by several types, ranging from questioning reports, transcribed confessions, land registry documents and telephone printouts. As a result, they exhibit many different phenomena that make them highly heterogeneous. A typical case is represented by some target pieces of information (e.g., the connection between people expressed by the relation KNOWS), which are not always realized by single sentences, but span much larger textual units (e.g. in a questioning report). Moreover, several extra-linguistic knowledge plays a role in establishing the correctness of some relations. For example, several criminal enterprises take their name from the place of origin and a systematic ambiguity arises.

In order to support the investigative analysis, the *REVEAL* project focused on the collection of living examples from real texts about the textual phenomena of interest for crime investigation, i.e. entity classes and specific relationships as those reported in Table 1. This phase corresponds to the analyst work. They are currently required to annotate every fact of interest directly on the target document, that is to collect *quotations*. In a machine learning perspective this corresponds to the development of training sets for the example-driven induction of a classification function. These keep track of the link between the semantic information to be extracted and the originating (host) text.

Relation	Description	Abbreviated Form
r_1	*A physical person knows another physical person*	PP KNOWS PP
r_2	*A physical person photographically identifies a physical person*	PP IDENTIFIES PP
r_3	*A physical person hangs out at a place*	PP HANGS OUT PL
r_4	*A physical person belongs to a criminal enterprise*	PP BELONGS TO CE
r_5	*A criminal enterprise includes a criminal enterprise*	CE INCLUDES CE
r_6	*A means of communication is linked to a juridical person*	MC IS LINKED TO JP
r_7	*A means of communication is linked to a physical person*	MC IS LINKED TO PP

Table 1: Relations Set

In the project, a specific manual annotation process has been thus designed. The machine learning team supported the analysts in the development of guidelines to focus on the textual aspects of the problem. In order for every undertaken decision about a document to be made reusable (e.g. for later training), the annotators have been asked to mark the exact text boundaries of their accepted relations, within each analysed document.

Several conceptual and linguistic problems emerged in this phase as a clear consequence of the high linguistic and semantic complexity, discussed hereafter.

Linguistic complexity. The natural language phenomena exhibited by the texts are highly heterogeneous. Most of the linguistic problems are related to the use of specific forms, as dialectal and jargon expressions, that open a variety of ambiguities to the interpretation, or to clerical errors during interrogations or audiotypings. This implies that the application of a syntactic parser is unhelpful as for coverage at the level of lexical and grammatical phenomena.

Moreover, a crucial problem is that interpretations are often open to subjectivity. Take, for example, a sentence like

Ne parlai con Mario e Giorgio [1]

that was treated differently by individual annotators. One accepted the relation KNOWS between the speaker and both entities `Mario` and `Giorgio`, and produced, in this way, three annotations for the three pairs of physical persons (PP): `(speaker,Giorgio)`, `(speaker,Mario)` and `(Giorgio,Mario)`. This interpretation clearly assumed that a meeting had taken place between the three. On the opposite, a second annotator outlined that no information could be found in the sentence confirming that the speaker met both persons at the same time. This alternative interpretation results into just two annotations between the speaker and one of the PP.

Consistency Although trained through very specific guidelines, annotators often show not to follow them strictly. This is largely due to the combinatorial explosion of some phenomena which are difficult to fully consider. This leaves some free space for the annotator to neglect some cases, thus reducing coverage. An example of such inconsistent behavior is the analysis of an excerpt like the following:

All'incontro a Roma erano presenti: Andrea, Barbara, Claudio, Daniela, Ettore e Francesca.[2]

It is obviously true that this sentence suggests binary relations between all pairs of the mentioned PP (hence, according to the annotation rules, we should have $6*(6-1)/2 = 15$ instances of the KNOWS relation), and between people and the location (i.e. Rome, with 6 HANG OUT relations between PPs and PLACE). One annotator pointed out for this sentence only the last 6 relations.

In order to handle the above problems, a quality test over the annotations has been carried out. An analysis of the results of two annotation teams working over the same set of documents has been carried out to assess the two deployed versions. In order to compare independent choices, various metrics have been applied, and the inter agreement between the two independent teams was measured. This analysis is discussed in section 3.1.

Within the above framework, the targeted relation extraction task can be thus formalized as follows. Let $\mathcal{O}$ and $\mathcal{R} = \mathcal{R}_{/2}$ denote the finite set of entity types and the binary relation types, respectively, and let t stands for a generic relevant fragment observable in a document. The task of recognizing a given relation $r \in \mathcal{R}$ for a text t_{ij}, including mentions to two entities e_i and e_j, whose types are $T_i, T_j \in \mathcal{O}$

[1] *I told it to Mario and Giorgio.*

[2] *Andrea, Barbara, Claudio, Daniela, Ettore and Francesca attended the meeting in Rome.*

respectively, formally corresponds to the function:

$$f(e_i, T_i, e_j, T_j, t_{ij}) \to \mathcal{R} \cup \{\perp\} \qquad (1)$$

where the special type $\perp$ is used to falsify all relations $r \in \mathcal{R}$. We will see in the next section how Eq. 1 can be mapped into a learning task over the set of annotation available, used as training examples.

2. AUTOMATIC RELATION EXTRACTION FOR INVESTIGATIVE TEXT ANALYSIS

The adoption of an empirical view on Relation Extraction from texts has been already studied within the machine learning community, as in [26, 10, 7], where variants of Support Vector Machines ([23]) are applied. The common idea of these works is that the computation of the function f (as in Eq. 1) is translated into an automatic classification step. The targeted entities e_i and e_j are here mapped into a vector $\vec{x}_{ij}$ of properties expressing different types of features of the text unit t_{ij} (i.e. a potential quotation) in which they appear. A boolean standpoint can be thus taken, where $f(e_i, T_i, e_j, T_j, t_{ij}) = r_k$ only when $H_k(\vec{x}_{ij}) = true$: in other words, the recognition is embodied by the hypothesis function $H_k(.)$, to be learnt, that accepts or rejects $\vec{x}_{ij}$ as an instance of the relation r_k. Functions $H_k(.)$ are binary classifiers for each relation r_k and can be acquired from existing repositories of annotated examples.

According to the risk minimization principle ([23]), SVM learning aims at finding the best classification function $H_k()$ able to separate negative from positive examples for a semantic relation r_k. During the classification stage, different SVMs are applied to a new example $\vec{x}_{ij}$ and the final multiclassification step selects the preferred class through the method known as *one vs. all* classification ([21]).

Support Vector Machines model the hypothesis function as an hyperplane $H(\vec{x}) = \vec{w} \times \vec{x} + b = 0$, where $\vec{x}$ is the feature vector representation of a source classifying text o whereas $\vec{w} \in \Re^n$ and $b \in \Re$ are parameters, learned from the training examples by applying the *Structural Risk Minimization principle* [24]. The object o is mapped in $\vec{x}$ with a feature function $\phi : \mathcal{O} \to \Re^n$, where $\mathcal{O}$ is the set of the objects that we want to classify. Hence o is categorized in the target class only if $H(\vec{x}) \geq 0$.

Kernel functions have received significant attention in this framework. SVM classifiers learn a decision boundary between two data classes that maximizes the minimum distance, or *margin*, of the training points of each class from the boundary. The adopted notion of distance and the feature space in which the boundary is set are determined by the choice of the kernel function ([22]). The kernel trick allows us to rewrite the decision hyperplane as:

$$\begin{aligned} H(\vec{x}) &= \Big(\sum_{i=1..l} y_i \alpha_i \vec{x}_i\Big) \cdot \vec{x} + b \\ &= \sum_{i=1..l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b = \sum_{i=1..l} y_i \alpha_i \phi(o_i) \cdot \phi(o) + b. \end{aligned}$$

where, y_i is equal to 1 for positive and -1 for negative examples, $\alpha_i \in \Re$ with $\alpha_i \geq 0$, o_i $\forall i \in \{1,..,l\}$ are the training instances and the product $K(o_i, o) = \langle \phi(o_i) \cdot \phi(o) \rangle$ is the kernel function associated with the mapping ϕ.

Note that, we do not strictly need to apply the mapping ϕ, as we can use $K(o_i, o)$ directly. This allows us, under the Mercer's conditions ([22]), to define abstract kernel functions that generate implicit feature spaces. The SVM optimization algorithm is guaranteed to converge to a global optimum that affords the geometric interpretation of margin maximization. The polynomial kernel gives an interesting example:

$$K_p(o_1, o_2) = (c + \vec{x}_1 \cdot \vec{x}_2)^d,$$

where c is a constant and d is the degree of the polynomial. This kernel generates the space of all conjunctions of feature groups up to d elements.

Besides these desirable properties, kernel methods have the advantages that combinations of kernel functions can be easily integrated into SVM as they are still kernels. The choice of the kernel can be also based on prior knowledge about the problem and on the noisy nature of the data. We can carry out two class of operations on the incoming data sets:

- kernel combinations, e.g. $K_1 + K_2$ or $K_1 \times K_2$
- feature mapping compositions, e.g. $K(o_1, o_2) = \langle \phi(o_1) \cdot \phi(o_2) \rangle = \langle \phi_B(\phi_A(o_1)) \cdot \phi_B(\phi_A(o_2)) \rangle$

Kernel combinations are very useful to mix the knowledge provided by the original features whereas these characterize feature spaces with quite different topological properties, for example acting on different perspectives (e.g. lexical vs. syntagmatic) on the original objects, e.g. textual units. Feature mapping compositions are thus useful methods to derive powerful kernel classes. In this work, we took advantage of the combination of two classes of kernel functions

In particular, a specific class of kernel function, i.e. the *string* (or *sequence*) *kernel* ([14]), already successfully applied to relation extraction tasks [26, 10, 7, 19, 15, 16], has been experimented. String kernels compute the similarity between instances according to their common sparse subsequences as observed in the targeted textual units t_{ij} and t'_{ij} used to represent them. Learning proceeds through the matching of subsequences as they are exhibited by training examples. During classification, a training and a test case are compared. Common sequences (with gaps) are efficiently matched according to dynamic programming techniques. A decay factor λ is imposed to lower the contributions to the overall score of those text portions characterized by longer gaps. In [7] the feature space is also pruned by structuring sequences of a quotation, according to their relative position with respect to the two involved entity mentions.

In our work a similar approach is used. We assume that a quotation here is intended as a text window that include the two entities. Usually, a structured representation in three segments is adopted. A *Fore-Between* segment (FB) is made by words in the sentence appearing between the n-th position before and the n-th position after the earlier entity in the text. The *Between* segment (B) is made of words that appear between the positions of the two entity mentions. Finally, the *Between-After* segment (BA) includes words appearing between the n-th position before and the

n-th position after the latter of the two entities in the text. These annotations (left-to-right FB to BA) are thus made available to match subsequences in the suitable positions relative to the entities. The quotation is usually considered the text span that covers the union of the FB, B and BA subsequences.

In the investigation domain, targeted here, relations of interest tend to be realized between entities even at a very long distance in the text. For this reason, as we will see in Section 2.1, we followed an approach simpler than the one in [7]. Only the two FB and BA sequences are considered as originating subsequences. A single sequence is then obtained through direct juxtaposition of FB and BA over which the kernel computation is run. Sometimes it covers the entire sentence where the entities e_i and e_j are both quoted. However, when longer distances and multiple sentences are treated, the resulting kernel acts only on text fragments, that are more local to e_i and e_j. As a result, this kernel is oriented to capture the shallow (local) syntactic information implicit in the fragments. It will be hereafter referred as K_{Seq}.

As kernel composition allows us to adopt several representations for an incoming text unit, we also exploited typical lexical representations of the source examples through a bag-of-word (BOW) approach. In this representation, analogously to the K_{Seq} representation, every token in a text window including the two entities (using also the n $Fore$ words of the entity earlier in the text and the n $After$ words of the latter entity) is taken into account for the kernel computation. The resulting kernel, K_{Bow} hereafter, is focused only on purely lexical data information and it does not allow to capture some task specific aspects, e.g., the distance between the involved entities. It has thus been extended through special features, as discussed in the next section: the resulting alternative model will be referred as K_{XBow}.

In this way, lexical and syntagmatic spaces are modelled independently, via K_{Bow} (or K_{XBow}) and K_{Seq} respectively. The overall kernel is defined through kernel combination. The usual sum has been here applied , i.e.
$K(X_1, X_2) = K_{XBow}(X_1, X_2) + K_{Seq}(X_1, X_2)$[3].

2.1 Feature Modeling for investigative relations

The adoption of an empirical perspective requires the availability of annotated examples of relationship instances as they are observed into incoming objects o (here the text units t_{ij}). Then individual o have been mapped into suitable vectorial forms $\vec{x}$. This step is carried out through the extraction of a set of properties (i.e., features) from the source objects t_{ij}.

Every analysts' annotation consists of a valid instance of a relationship class $r \in \mathcal{R}$. These are gathered as the set of positive training examples for the relation r. Moreover, every positive instance for a relation, say r_k, is also a negative example for every relation r_l $(l \neq k)$ that insists on the same entity pairs of r_k: for example, every accepted instance of the KNOWS relation (between PP pairs) is also a negative instances for the IDENTIFIES relation (see Table 1 for a full description). However, negative training examples also stem from rejected cases. In order to build the full set of negative examples, we computed all possible entity pairs from a document that: (1) are not positive examples of any relation, and (2) obey to at least one relationship class in the domain schema (i.e., it is an entry in Table 1). This assumption states that every candidate quotation, suggested by at least one candidate entity pair, is a negative example when no annotation is available for it.
Notice that the above assumption make the set of candidate pairs to proliferate in long documents. However, *inter-sentence* relations between very far sentences are very infrequent. In order to keep manageable the candidate pair set, thresholds to the maximal distance allowed between two entities e_i and e_j are imposed. The analysis of the annotated corpus showed that most of the entity pairs in valid relation instances generally occurred within a limited distance[4] The distribution of valid relations allowed us to define a criteria (statistical filter hereafter) that filter out the (e_i, e_j) pairs whose distance is above a threshold. The optimal threshold has been estimated over a development set as the 90-th percentile that maximizes coverage while minimizing the number of false instances introduced. As different relations produce different distributions different thresholds have been adopted for each relationship class. The statistical filter is then clearly applied in the training (to gather useful negative examples) as well as in the test phase.

The complexity of the relation extraction task targeted in this project asks for a suitable (vector) description $\vec{x}_{ij}$ of individual examples t_{ij}. Features have to cover a variety of phenomena ranging from lexical information (e.g., expressing the main verbs denoting the target relations, such as *to_meet* for relation KNOWS) to grammatical constraints. Moreover, task specific features have been designed to better capture textual hints. In all the experiments the following set of features has been adopted.

Lexical units. Words in texts are expressed through their surface representations (tokens) or through the corresponding lemmatized forms (lemmas)[5].

Entity Types. In order to increase the generalization power of individual features, the textual mentions to entities (e.g. "*Mario*", "*Roma*") are substituted by the labels of their corresponding class. For example in the excerpt *"Lui ha abitato a Roma per un periodo*[6]", the active tokens in the representation become {PP, *ha, abitato, a*, PL, *per, un, periodo* }.

Distance between mentions to entities. Although the token distance between the involved entities is used as a filter for candidate pairs, the distance is also useful to impose more or less stricter criteria on other features. So, discrete values are obtained as the 3 main percentile (33%, 66%, 100%) of the distributions of distance values over the set of individual relation instances. As a result different three-valued labeling are obtained for different relationship classes.

Punctuation. Punctuation in the *Fore*, *Between* and *After*

[3] A normalized version $K_{Norm}(X_1, X_2)$ is adopted for all the kernels K, where $K_{Norm}(X_1, X_2) = \frac{K(X_1,X_2)}{K(X_1,X_1)K(X_2,X_2)}$

[4] Distance is measured in term of number of tokens.

[5] Early work on the use of syntax for text categorization suggests that part-of-speech tags, e.g. [3], can be used as a simple and powerful features.

[6] *He has been living in Rome for a while.*

portions of the involved textual units t_{ij} are all represented via special labels, accounting for the relative position of each punctuation mark with respect to the entities. For example, a comma in the *Fore* component of a textual unit (i.e., before the entity e_i appearing earlier in the text) is denoted by `#,F`, while `#,B` is reserved for commas appearing between the two entity mentions. Moreover, each feature is weighted according through its number of occurrences within the corresponding component (e.g., *Fore* vs. *Between*).

Ordering of mentions. This boolean feature *Ord* denotes the property of the textual unit t_{ij} to instantiate a relation r_k in agreement with the order of this latter. For example, while the HANGS OUT relation is clearly orientated from people PP to places PL, the fragment "*A Roma l'incontro con Mario si protrasse sino a tarda notte*[7]" mentions the two entities in the reverse order: in this case, the feature *Ord* assumes the value `false`.

3. PERFORMANCE EVALUATION

The industrial impact of the proposed SVM-based technology has been evaluated on real test collections, in coordination with the analyst teams. The overall objective of the experiments was to assess the quality of the datasets, to provide a comparative analysis of different learning algorithms, and measure the accuracy reachable. Some aspects more related to the applicability of REVEAL to the current operational investigative practices are discussed in the conclusive Section 4. This section first discusses the experimental set-up (Section 3.1), this including the analysis of the seeding annotated corpus. In Section 3.2 the comparison of different learning algorithms over the employed test data. Finally, an analysis of the role of individual kernels for the different relations is reported in Section 3.3.

3.1 Experimental Set-up

The experimental corpus, made of 86 documents, annotated by two teams of analysts, has been extracted from two collections of public judicial acts related to the legal proceedings against the same large criminal enterprise. The corpus has been split into a 90% component (i.e., 79 documents) for training and the remaining 10% (7 documents) then used for testing[8]. The splitting has been applied by trying to preserve, in the test set, the same distribution of instances across relationship classes as those observed in the training data. Although manual annotations have been added for 15 different relationship classes, due to lack of evidence in the training data for some classes, the experimentation was focused on the seven relations reported in Table 1. Skewed distributions are observed, where some relations are much more common in documents like PP HANGS OUT AT A PL or PP KNOWS PP and other are very infrequent as ASSET IS CONNECTED TO A PLACE. Some of the relations, although high relevant for investigation, were not well represented in the training data. This allows us also to better verify the robustness of the REVEAL models. The experimental corpus is described in Table 2. It shows the overall number of instances available for training (column 2) and testing (column 3) over each individual relation: percentages are relative to the number of positive cases that have been used for training. Notice how the first two rows (relations KNOWS and IDENTIFIES) have the same number of cases: they in fact operate on the same number of candidate pairs, as their semantic signature (i.e., $(PP \times PP)$) coincides.

Id	Relationship Class	Training instances (% of positives)	Test instances
r1	PP KNOWS PP	3985 (16.18%)	519
r2	PP IDENTIFIES PP	3985 (5%)	519
r3	PP HANGS OUT PL	2359 (14.83%)	229
r4	PP BELONGS TO CE	1717 (35.11%)	103
r5	CE INCLUDES CE	604 (20.19%)	10
r6	MC IS LINKED TO JP	62 (51.6%)	22
r7	MC IS LINKED TO PP	231 (42.85%)	39

Table 2: Experimental Data Set

As discussed in Section 1.1, some complex problems afflicts the annotation phase. In order to evaluate the quality of the annotated material produced by the analysts as well as for evaluating the consistency of the test material an inter-annotator agreement, a test has been performed. All the 7 test documents have been annotated by a second team (made of analysts not included in the first one), which was trained according to the same modalities of the first team. After a short training on separate documents they replicated the decisions so that all test cases have been doubly annotated. The measure of the inter-annotator agreement observable between the two teams was the *Cohen's Kappa* ([8]), computed as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)},$$

where $P(A)$ is the observed agreement among raters of the two teams, and $P(E)$ is the expected agreement, that is, the probability the raters agree by chance. The values of κ lie within the interval $[-1, 1]$: $\kappa = 1$ means that the raters are in perfect agreement, $\kappa = 0$ that they agree by chance while $\kappa = -1$ expresses total disagreement[9]

The inter-annotator agreement measures are reported Table 3 according to κ values for each individual relation. The inter-annotator agreement measures are reported in Table 3 according to κ values for each individual relation. As expected, high κ values are obtained for almost all relations. However, although prevalence suggests not to emphasize the criticality of values lower than 65%, the annotations of relation r_1 (i.e. KNOWS) are much controversial between the teams. This is due to its combinatorial nature, already outlined in Section 2.1, that implies a large number of diverging choices or missing cases (for both teams). Notice that relation r_1 is also the most likely in the data sets (see Table 4) so that the overall κ is quite low (i.e. 54,44%). This

[7] *"In Rome, the meeting with Mario lasted 'til late night"*

[8] A distinct set of 10 documents has been used as a development set to optimize the parameter settings for all the compared algorithms.

[9] As discussed in [11] there are two ways to estimate $P(E)$. In our cases, where 2 raters (indexed through i) and 2 categories (indexed by j) are involved, $p_{i,j}$ denotes the probability that rater i accept the j-th case. Then, an estimate $P(E) = p_{1,1} * p_{2,1} + p_{1,2} * p_{2,2}$ has been adopted. This implies that κ is affected both by the *bias* and *prevalence* problems. While we cannot avoid the bias problem, prevalence must be taken into account for interpreting the test outcomes.

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	Overall
Candidate pairs	10.839	10.839	2.182	1.441	264	256	720	26.541
Pairs accepted by team 1	56	9	53	51	3	7	10	189
Pairs accepted by team 2	330	10	80	54	4	6	10	494
Y1 & Y2	56	9	53	50	3	6	10	187
Y1 & N2	0	0	0	0	0	1	0	1
N1 & Y2	274	1	27	4	1	0	0	307
N1 & N2	10.509	10.829	2.102	1.387	260	249	710	26.046
Cohen's κ (%)	28.38%	94.73%	79.09%	96.05%	85.53%	92.11%	100%	54.44%

Table 3: Inter-annotator agreement according to *Cohen's Kappa*

		Team 1	
		Accepted	Rejected
Team 2	Accepted	56	274
	Rejected	0	10509

Table 4: Confusion matrix for relation r_1.

confirms the complexity of the targeted relation extraction task even for expert analysts. Although the inter-annotator agreement is rather low, in all the test discussed in this section and for every relation, a case is considered positive for a relation *if* almost a team has accepted it. For this reason most of the performance scores discussed in the next section can be considered lower bounds to the quality reachable via a ML approach.

3.2 Comparative Analysis

A second set of experiments was run in order to compare different learning approaches on the available experimental data sets. While the first experiment confirmed the high complexity of the targeted task, in a second experimental stage we wanted to evaluate the impact of different feature models across a set of learning strategies. In order to test the impact of the REVEAL models against some performance baselines, we adopted two well-known learning algorithms, i.e. C4.5 decision tree learner[20] and a NaiveBayes model to the same data sets[10]. Both systems have been run over the feature set characterizing the K_{XBOW} kernel (i.e., bag-of-words extended with the domain features discussed in Section 2.1). Moreover, a simple baseline making random choices across the candidate pairs (filtered according to the 90-th percentile statistics), has been evaluated. All the algorithms were optimized over the same development set and then tested against the data shown in Table 2. For evaluation, the classical evaluation metrics have been used. Precision (P) (i.e. the percentage of correctly recognized relation instances against the total number of accepted test cases), recall (R, i.e., the percentage of correctly recognized relation instances against the total number of true relationship instances present in the test documents) and the F-measure (F1), as the harmonic mean between precision and recall (with equal balancing among the two). Micro average is used to summarize the results of individual relations. Accuracy has been also measured as the percentage of correct recognition inferences, this including the acceptance of correct candidates and the rejection of false candidates.

Algorithm	P	R	F1	Acc
Random Choice	0.13	0.4	0.21	41%
Decision Tree	0.45	0.24	0.31	54%
NaiveBayes	0.34	0.56	0.40	57%
K_{BOW}	0.32	0.75	0.45	66%
K_{XBOW}	0.70	0.83	0.73	85%
$K_{XBOW} + K_{Seq}$	0.75	0.85	**0.75**	**88%**

Table 5: Comparative evaluation among classification algorithms

The comparative evaluation is shown in Table 5 where the performances obtained by the best parametrization of the different algorithms are shown. The last three rows represent the systems trained over the different kernels used by REVEAL[11]. Although the precision score of Decision Tree and NaiveBayes are better than the model trained over the bag-of-words (i.e. a simple model), it achieves an overall lower F1 measure (0.31 and 0.4 vs. 0.45) this is due to the higher generalisation power of the kernel methods, in fact the simple Bow model is already able to achieve an higher recall level. The SVM models all show a good coverage with a recall scores over 0.75. The REVEAL two models (i.e. K_{XBOW} and $K_{XBOW} + K_{Seq}$) are the best performing models.

3.3 Feature Analysis

The good results obtained through the different kernels, as shown by Table 5, inspired an analysis of the impact of the different models over the individual relations. As discussed in Section 2.1, the extended features that characterize some conceptual and task specific properties of the individual text units t_{ij} are used to augment the kernel expressiveness and generalization power. This is shown by the extension of the *bow* model through the XBOW one.

Notice how the extended features have several variants that imply several learning configurations to be evaluated. For example, lemmas and tokens can be used, and conceptual labels can be adopted to generalize the names of entity instances. In order to find the best variants several tests have been run. The best trade-off between precision and recall scores was achieved with the following feature configuration:

[10] Both algorithms have been tested through Weka ([25]).

[11] For the SVM learning, we used the SVMlightTK platform as available at: `http://dit.unitn.it/ moschitt/Tree-Kernel.htm`. The sequence kernel supported by that platform is obtained as a special case of the tree kernel, as discussed for example in [18].

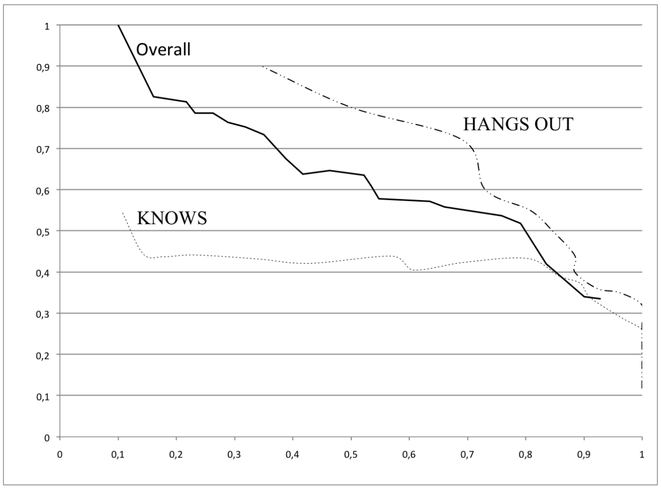

Figure 1: Precision/Recall curve

- *Lexical Units*: tokens
- *Entity Types*: textual mentions to entities ($Mario$) are substituted with their corresponding type labels (PP) in all representations (even in the sequence kernel structures FB and BA)
- *Distance*: number of tokens between the two involved entities
- *Punctuation*: expressed only for marks appearing *between* the two entities: other marks are neglected from the analysis
- *Ordering of mentions*: applied as boolean feature

In Table 6 the F-measure scores as obtained for individual relations according to the above XBOW model are reported. Most of the relations obtain an excellent score, reaching in some case an F1 of 1. On some more complex relationship classes, as PP KNOWS PP and PP HANGS OUT *at a* PL, the K_{XBOW} kernel achieves lower performances, basically due to the presence of dialectal or syntactically odd expressions. The combination of the two kernels, last column of Table 6 seems to overcome most of these problems. Notice that the weaker relation is r_1 (KNOWS) where also experts show a very high disagreement. It seems that, although relatively shallow features are adopted and no syntactic parsing is applied, the trained SVM performs on most of the phenomena similarly to humans: relation detection exhibits a similar behavior where complex cases are hard for both. In particular, if the $K_{XBOW} + K_{Seq}$ kernel is only applied to the 335 cases (that is the 65% of the overall test set) where full agreement among the annotator teams is observed, its F1 achieves the much better value of 0.82 (vs. 52 %).

As a final test, we computed the precision-recall curve for the REVEAL kernel $K_{XBOW} + K_{Seq}$, obtained according to different parameter settings (controlling the trade-off between recall and precision). The curve, reported in Figure 1, compares the behavior of the system over two relationship classes with the micro-averaged results over all relations (*overall*). As apparent, the plot shows a regular shape and it suggests that parameter tuning can be effectively applied to capture the required trade-off between the suitable coverage and the required accuracy of the method. Notice that optimizing coverage can be a much more critical requirement within the investigative domain.

Id	Relationship Class	K_{XBOW}	$K_{XBOW} + K_{SK}$
r1	PP KNOWS PP	0.398	0.523
r2	PP IDENTIFIES PP	1	1
r3	PP HANGS OUT at a PL	0.40	0.684
r4	PP BELONGS to CE	0.66	0.747
r5	CE INCLUDES CE	1	1
r6	MC IS LINKED TO JP	0.70	0.70
r7	MC IS LINKED TO PP	1	1

Table 6: F-measure score of the SVM models over individual relationship classes

4. CONCLUSIVE REMARKS

The technology designed and tested in the project has been shown to be very effective. Outcomes can be discussed with respect to benefits at the *process level* and at the *technological level.*

In the first view, the afforded empirical perspective has allowed us to approach a very complex task within a novel workflow that is highly innovative with respect to current practices. The engineering criteria followed in REVEAL were used to verify the quality and consistency of existing procedures. The analysis process proposed in the project has several beneficial effects in the productive line of the targeted domain. First, it will allow us to capitalize the huge amount of textual document by storage, indexing and maintenance of crucial information covering several semantic phenomena. Their linking to originating texts enables a variety of future uses paving the way to sophisticated forms of technology-supported investigation. Automatic extraction of entities and relations from texts is here considered a first step towards a deeper semantic approach to the overall investigation workflow. The noisy nature of investigation data takes advantage from kernel methods, in particular from the sequence kernel computation that allows to learn some specific lexical and syntactical domain phenomena.

Several technological benefits are related to the specific modeling proposed in REVEAL, as discussed in Section 2. A wide range of experimental activities have been discussed in the paper (Section 3). As a result, the automation of the annotation step cannot be yet considered as a comprehensive solution, as performances are not always acceptable. However, a very attractive semi-automated solution has been enabled where the analysts role is to inspect the system suggestions to validate them. As validation is quite simpler than annotating "*from scratch*", a significant overall speed-up can be expected. During the project, the REVEAL processing time for a medium sized document of 15 pages has been estimated being about 13 minutes[12]. The current professionals are able to annotate on average the same amount of

[12] A modern dual core workstation was employed for our computation: better efficiency can be obtained by means of more powerful platforms.

text in not less than 4 hours. Although the latter measure is surprisingly good, given the complexity of the task, the resulting speed-up, of a straightforward application of REVEAL to the same task, is about 18 times. For a longer document (about 300 pages), that analysts annotate in 10 days, the overall processing time of REVEAL is 4 hours. The speed-up here is about 60. The impressive impact that the above issues can have on the productivity of the analyst teams will be part of future evaluation studies aiming to better quantify it.

A final remark is about the quality achievable by an automatic approach. As this study confirms, often the human analysis of noisy data has been shown critically error prone, this resulting in missing or inconsistent information. To our knowledge, no analytical measures for such errors have been previously carried out in the targeted organizations. In some sense, awareness about the quality of most of the stored information, made available to the investigators, is very poor. One of the beneficial effects of the project has been to raise these questions within the cooperating organizations (i.e. technology providers and investigative agencies). Part of the future work will be thus to analyse quantitatively these aspects and frame them within a wider view on these semantics-enabled investigation technologies.

Regarding future enhancement of our system, we would like to exploit advanced shallow semantic approaches such as predicate argument structures, e.g. [12, 17, 13, 18]. Additionally, term similarity kernels, e.g. [2, 4], will be likely improve relation generalization, especially when combined syntactic and semantic kernels are used, i.e. [5, 6].

5. REFERENCES

[1] Astrea, information and communication for justice. coordinated by Italian Research Council/Research Institute on Judicial Systems (IRSIG-CNR), URL: `http://astrea.cineca.it/`.

[2] R. Basili, M. Cammisa, and A. Moschitti. Effective use of WordNet semantics via kernel-based learning. In *Proceedings of CoNLL-2005*, Ann Arbor, Michigan, 2005.

[3] R. Basili, A. Moschitti, and M. T. Pazienza. A text classifier based on linguistic processing. In *Proceedings of IJCAI 99, Machine Learning for Information Filtering*, 1999.

[4] S. Bloehdorn, R. Basili, M. Cammisa, and A. Moschitti. Semantic kernels for text classification based on topological measures of feature similarity. In *Proceedings of ICDM 06, Hong Kong, 2006*, 2006.

[5] S. Bloehdorn and A. Moschitti. Combined syntactic and semantic kernels for text classification. In *Proceedings of ECIR 2007, Rome, Italy*, 2007.

[6] S. Bloehdorn and A. Moschitti. Structure and semantics for expressive text kernels. In *In proceedings of CIKM '07*, 2007.

[7] R. Bunescu and R. Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 171–178. MIT Press, Cambridge, MA, 2006.

[8] J. Carletta. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22(2):249–254, 1996.

[9] X. Carreras and L. Marquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proc. of CoNLL 2005, Ann Arbor, Michigan*, pages 152–164, 2005.

[10] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of ACL'04*, pages 423–429, Barcelona, Spain, 2004.

[11] B. Di Eugenio and M. Glass. The kappa statistic: a second look. *Comput. Linguist.*, 30(1):95–101, 2004.

[12] A.-M. Giuglea and A. Moschitti. Knowledge Discovery using Framenet, Verbnet and Propbank. In A. Meyers, editor, *Workshop on Ontology and Knowledge Discovering at ECML 2004*, Pisa, Italy, 2004.

[13] A.-M. Giuglea and A. Moschitti. Semantic Role Labeling via Framenet, Verbnet and Propbank. In *Proceedings of ACL 2006*, Sydney, Australia, 2006.

[14] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[15] A. Moschitti. Kernel methods, syntax and semantics for relational text categorization. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 253–262, New York, NY, USA, 2008. ACM.

[16] A. Moschitti. Syntactic and semantic kernels for short text pair categorization. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 576–584, Athens, Greece, March 2009. Association for Computational Linguistics.

[17] A. Moschitti and A. B. Cosmin. A semantic kernel for predicate argument classification. In *CoNLL-2004*, Boston, MA, USA, 2004.

[18] A. Moschitti, D. Pighin, and R. Basili. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, Special Issue on Semantic Role Labeling(3):245–288, 2008.

[19] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *Proceedings of ACL'07*, Prague, Czech Republic, 2007.

[20] R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[21] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[22] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[23] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[24] V. N. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, 1998.

[25] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/˜ml/weka.

[26] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, 2003.