Kernel-based Relation Extraction for Crime Investigation

Roberto Basili[1], Cristina Giannone[2], Chiara Del Vescovo[2], Alessandro Moschitti[3], and Paolo Naggar[2]

[1] University of Roma, Tor Vergata, Rome, Italy
basili@info.uniroma2.it,
[2] CM Sistemi s.p.a., Rome, Italy
{cristina.giannone,chiara.delvescovo,paolo.naggar}@gruppocm.it,
[3] University of Trento, Trento, Italy
moschitti@disi.unitn.it

Abstract. In a specific process of business intelligence, i.e. investigation on organised crime, empirical language processing technologies can play a crucial role. The analysis of transcriptions on investigative activities, such as police interrogatory, for the recognition and storage of complex relations among people and locations 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 realised, 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 behaviour 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). Currently the system population process is executed by teams of analysts that, reading each document, annotate all quotations about facts and involved subjects according to a conceptual schema. Notice that the analysis process involves two main phases:
Analysis phase: a document is analyzed for the first time by an analyst that gathers the information into a knowledge base.

Validation phase: a second analysis process is applied to verify data from the first phase. This allows to correct data lacks, errors and flimsiness.
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.
The huge amount of texts involved in the investigation process asks for the automation of the recognition of the specific relations. 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 as shown in international challenges ([2]). Second, the adoption of inductive methods enables an incremental approach where the interleaving between the automatic learning for tagging relations and human validation allows to scale up in a much cheaper fashion: at the i-th iteration, professional analysts may more quickly provides novel annotations through the corrections of mistakes made by the system, and this triggers a novel stage $(i+1)$ with a larger training datasets. Moreover, annotations provide links between relational facts and their textual realisations that are very important in the long term perspective. This is particularly true in the adoption of an empirical view in the management of large archives. On the contrary, links are currently practices used only during the preliminary analysis phase to justify the final decision to updating the underlying information system: no trace is left of the textual anchoring of the stored facts. While this allows to share structured information among investigative working groups distributed across the country, this does not allow to capitalise the analysis phase in future activities. This work is aimed at presenting the main results of the *REVEAL* project and, in particular, how the prototype can be used to support this view. As we will see, the proposed methodology has been applied successfully in the project and early results confirm the wide applicability of the proposed approach. The project aims to substitute the manual *analysis phase* employing REVEAL system for real-time extraction of information. This paper reports the first experiences in using the system and a comprehensive set of validative results.

Rel Id	Description	Abbreviated Form	Rel Id	Description	Abbreviated Form
r_1	*A physical person knows another physical person*	PP KNOWS PP	r_4	*A physical person belongs to a criminal enterprise*	PP BELONGS TO CE
r_2	*A physical person photographically identifies a physical person*	PP IDENTIFIES PP	r_5	*A criminal enterprise includes a criminal enterprise*	CE INCLUDES CE
r_3	*A physical person hangs out at a place*	PP HANGS OUT PL	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

1.1 Definition of the task

Text mining for the crime investigation domain is a complex semantic task where textual, linguistic and domain knwoledge are critically involved. Relation extrac-

tion is at the cross-road of all these knowledge sources. Typical textual phenomena of interest for crime investigation refer to classes of entities and specific relationships, as those reported in Table 1, have been choose to carried out the experimental activities. The set of documents relevant for the peculiar analysis in this domain are of different types like questioning reports or transcribed confessions, land registry documents or telephone printouts. As result, they exhibit many different phenomena that make them semantically and syntactically highly heterogeneous. A typical case is represented by some target information (e.g. the connection between people expressed by the relation KNOWS), that is not always realized within single sentences, but can span much larger textual units. Moreover, several extra-linguistic information plays a role in establishing the correctness of some relation. 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 focuses on the collection of living examples from real texts. This phase corresponds to the first phase of analysts' work. They are currently required to annotate every fact of interest directly on the target document, that is to collect *quotations*. In machine learning, this corresponds to create training examples for automating the analysis phase, that is keeping track of the link between the target structured information to be extracted and the originating (host) text.

This manual annotation phase has thus been designed in the project. The machine learning team of the project supported the analysts in the development of guidelines to focus on the textual aspects of the problem. In order to collect instances for the training phase, 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 complexity of documents as discuss hereafter:

Linguistic complexity. The natural language phenomena present in the text 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. This renders the application of a syntactic parser very problematic as for coverage at the level of lexical and grammatical phenomena. A crucial problem is that interpretations are often open to subjectivity. Take, for example, a sentence like

Ne parlai con Mario e Giorgio [4]

that was differently annotated 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 subjects at the same time. This alternative interpretation results into just two annotations between the

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

speaker and each PP.
Consistency Although trained about very specific rules and guidelines, annotators often show not to follow them strictly. This is largely due to the combinatorial explosion of some phenomena that is difficult to be pointed out completely. This leaves some free space for the annotators to neglect some cases, thus reducing coverage. An example of such inconsistent behaviour is the analysis of an excerpt like the following:

All'incontro a Roma erano presenti: Andrea, Barbara, Claudio, Daniela, Ettore e Francesca.[5]

It is obviously true that this sentence suggests binary relations (KNOWS relation) between all pairs of the mentioned physical persons, and between people and the location (i.e. Rome, with 6 HANGS 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 study of the quality of the annotations has been carried out. Two annotation teams have been employed over the same documents set to measure the quality of the two deployed versions. Various metrics have been applied to compare independent choices, and the inter agreement among 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 formalised as follows. Given a finite set of entity types $\mathcal{O}$ and binary relation types $\mathcal{R} = \mathcal{R}_{/2}$, and any relevant fragment t observable in a document, the task of recognising 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 $T_i, T_j \in \mathcal{O}$ respectively, formally corresponds to the function:

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

where the special type $\perp$ is used to falsify all relations $r_i \in \mathcal{R}$. We will see in the next section how the 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 [3–5], where variants of Support Vector Machines ([6]) 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 $\boldsymbol{x}_{ij}$ 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 $H_k(\boldsymbol{x}_{ij}) = true$: in

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

other words, the recognition is embodied by the hypothesis function $H_k(.)$, to be learnt, that accepts or rejects $\boldsymbol{x}_{ij}$ 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.

SVM classifiers learn a decision boundary between two data classes that maximises the minimum distance, or *margin*, from the training points of each class to 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 ([7]).

These methods have the advantage that combinations of kernel functions can be easily integrated into SVM as they are still kernels. Kernel combinations are very useful to mix the knowledge provided by the original features whereas these characterise 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.

A particular class of kernel function, successfully applied to relation extraction tasks [3–5], is the *string* (or *sequence*) *kernel* one ([8]). String kernels compute the similarity between instances according to their common sparse subsequences as observed in the targeted textual units t_{ij} used to represent them. Learning proceeds through the matching of such subsequences as they are exhibited by training examples. During classification, 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 characterised by longer gaps.

Analogously in our work we used a similar structuring. A quotation here is intended as a text window that includes the two target 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) include 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 cover 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 [5]. 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 results, 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 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, 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 in the BOW. The resulting kernel, K_{BOW} hereafter, cannot 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 modeled independently, via K_{BOW}/K_{XBow} and K_{Seq} respectively. The overall kernel is defined through the follow kernel combination: $K(X_1, X_2) = K_{XBow}(X_1, X_2) + K_{Seq}(X_1, X_2)$[6].

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 are mapped into suitable vectorial forms $\boldsymbol{x}$. 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 is used to build the set of positive training examples for the relationship class $r \in \mathcal{R}$. 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 instance for the IDENTIFIES relation (see Table 1 for a full description). In addiction, 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 are very infrequent between very far sentences. 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[7] The distribution of valid relations allowed to define a criteria (statistical filter

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

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

hereafter) that filters 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 maximises coverage while minimising the number of false instances introduced. As different relations produce different distributions different thresholds are 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 $\boldsymbol{x}_{ij}$ 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. Other, task specific features have been designed to better capture textual hints. In all the experiments the set of features described below has been adopted.

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

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*[8]", 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 percentiles (33%, 66%, 100%) of the distributions of distance values for each individual relations. Different three-valued labeling are obtained for different relationship classes.

Punctuation. Punctuation in the *Fore*, *Between* and *After* 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 OM 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*[9]" mentions the two entities in the reverse order: in this case, the feature OM assumes the value `false`.

[8] *He has been living in Rome for a while.*
[9] *"In Rome, the meeting with Mario lasted 'til late night"*

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 annotated corpus. In Section 3.2 the comparison of different learning algorithms over the employed test data.

3.1 Experimental Set-up

The experimental corpus, made of 86 documents, annotated by two teams of analysts, has been derived 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[10]. 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 AN ASSET IS CONNECTED TO A PLACE. Some of the relations were not well represented in the training data but they have been selected for their high relevance for investigations. This allows also to 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.

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 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

[10] A distinct set of 10 documents has been used as a development set to compute the parameters, such as filters and SVM setting

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

cases have been doubly annotated. The measure of the inter-annotator agreement observable between the two teams was the *Cohen's Kappa* ([9]), 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 $[-1, 1]$: This values mean, respectively, total disagreement and perfect agreement between the raters.

The inter-annotator agreement measures are reported in Table 3 according to κ values for each individual relation.

	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*

As expected, high κ values are obtained for almost all relations. However, although prevalence suggests not to emphasise 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). This confirms the complexity of the targeted relation extraction task even for expert analysts. Although the slightly low agreement, in all the test discussed in this section and for every relation, a case is considered positive for a relation *only if* both teams have 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 experimental data sets available. 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[10] and a NaiveBayes model to the same data sets[11]. Both systems have been run over the feature set characterising 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 tested against the data shown in Table 2. The comparative evaluation is shown in Table 4 where performances of the different algorithms are shown. The last three rows represent kernel models trained over the different kernels used by REVEAL[12]. Although precision of NaiveBayes is better than the model trained over the bag-of-words (i.e. a simple model), it achieves an overall lower F1 measure (0.39 vs. 0.45). 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 model.

Algorithm	P	R	F1	Acc
Random Choice	0.13	0.4	0.21	41%
Decision Tree	0.21	0.26	0.23	66%
NaiveBayes	0.36	0.48	0.39	63%
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 4. Comparative evaluation among classification algorithms

The good results obtained through the different kernels, as shown by Table 4, inspired an impact analysis of the different models over the individual relations.

In Table 5 the F-measure scores as obtained for individual relations according to the REVEAL models are reported. Most of the relations obtained 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 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 5, seems to overcame most of these problems. Notice that the weaker relation is

[11] Both algorithms have been tested through Weka ([11]) according to its standard parameter settings.

[12] 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 [12].

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 5. F-measure score of the SVM models over individual relationship classes

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 seems to deal with most of the phenomena in an harmonic way with humans: relation detection exhibits a similar behaviour where complex cases are hard for both. In particular, if the is $K_{XBOW} + K_{Seq}$ kernel is applied only to the 755 cases (that is the 65% of the overall test set) where full agreement is observed, its F1 achieves the much better value of 0.82.
As a final test, we computed the precision-recall curve for the REVEAL kernel $K_{XBOW} + K_{Seq}$, obtained by varying the SVM parameters and shifting the hyper-plane. The curve, reported in Figure 1, defines the trade-off between recall and precision for two relationship classes and the micro-averaged results from 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 accuracy of the method.

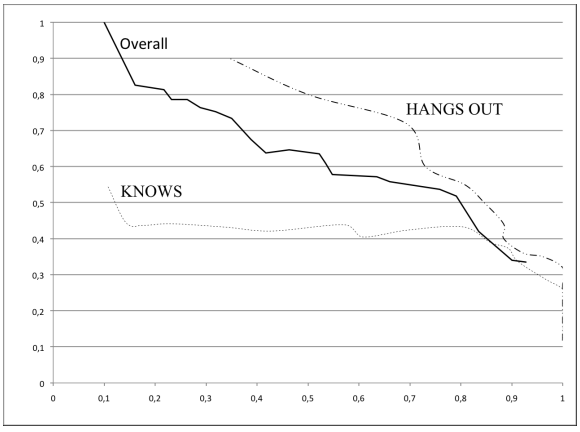

Fig. 1. Precision/Recall curve

4 Conclusive Remarks

From an industrial point of view the REVEAL project represented a relevant and successfully experience. 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 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 to capitalise 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.

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 results, 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[13]. The current professionals are able to annotate on average the same amount of text in not less than 4 hours. Although this 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 (300 pages), that analysts annotate in 10 days, the overall processing time of REVEAL is 4 hours. The speed-up here is about 60. A final remark is about the quality achievable by an automatic approach. As this study confirms, often human analysis has been shown critically error prone, this resulting in missing or inconsistent information. To our knowledge of the domain, no analytical measures for such errors have been previously carried out in the targeted organisations. In some sense, awareness about the quality of most of the stored information, made available to the investigators, is very poor. Part of the future work will be thus to analyse quantitatively these aspects and frame them within a wider view on these class of semantics-enabled investigation technologies.

[13] A traditional dual core Workstation has been employed for these measures: an even better efficiency can be obtained over more powerful infrastructures.

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. Carreras, X., Marquez, L.: Introduction to the conll-2005 shared task: Semantic role labeling. In: Proc. of CoNLL 2005, Ann Arbor, Michigan. (2005) 152–164
3. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3** (2003) 1083–1106
4. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of ACL'04, Barcelona, Spain (2004) 423–429
5. Bunescu, R., Mooney, R.: Subsequence kernels for relation extraction. In Weiss, Y., Schölkopf, B., Platt, J., eds.: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA (2006) 171–178
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
7. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
8. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. Journal of Machine Learning Research **2** (2002) 419–444
9. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Comput. Linguist. **22**(2) (1996) 249–254
10. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
11. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)
12. Moschitti, A., Pighin, D., Basili, R.: Tree Kernels for Semantic Role Labeling. Computational Linguistics **Special Issue on Semantic Role Labeling**(3) (2008) 245–288