IJDAR (2011) 14:213–228
DOI 10.1007/s10032-010-0138-0

ORIGINAL PAPER

Supervised semantic relation mining from linguistically noisy text documents

Cristina Giannone · Roberto Basili · Paolo Naggar · Alessandro Moschitti

Received: 15 December 2009 / Revised: 6 August 2010 / Accepted: 19 October 2010 / Published online: 16 November 2010

Abstract In this paper, we present models for mining text relations between named entities, which can deal with data highly affected by linguistic noise. Our models are made robust by: (a) the exploitation of state-of-the-art statistical algorithms such as support vector machines (SVMs) along with effective and versatile pattern mining methods, e.g. word sequence kernels; (b) the design of specific features capable of capturing long distance relationships; and (c) the use of domain prior knowledge in the form of ontological constraints, e.g. bounds on the type of relation arguments given by the semantic categories of the involved entities. This property allows for keeping small the training data required by SVMs and consequently lowering the system design costs. We empirically tested our hybrid model in the very complex domain of business intelligence, where the textual data are constituted by reports on investigations into criminal enterprises based on police interrogatory reports, electronic eavesdropping and wiretaps. The target relations are typically established between entities, as they are mentioned in these information sources. The experiments on mining such relations show that our approach with small training data is robust to non-conventional languages as dialects, jargon expressions or coded words typically contained in such text.

C. Giannone (✉) · P. Naggar
CM Sistemi s.p.a, Rome, Italy
e-mail: cristina.giannone@gruppocm.it;
giannone@info.uniroma2.it

P. Naggar
e-mail: paolo.naggar@gruppocm.it

C. Giannone · R. Basili
University of Roma, Tor Vergata, Rome, Italy

R. Basili
e-mail: basili@info.uniroma2.it

A. Moschitti
University of Trento, Trento, Italy
e-mail: moschitti@disi.unitn.it

1 Introduction

Mining relational patterns is a core activity of data mining testified by important previous work, e.g. [16,17,48]. From human being perspective, such research assumes a particular interest when the involved data are natural language documents and the relationships are defined between entities described in text, e.g. [23,24,34,47].

The relation extraction (RE) from text has been standardized by the automatic content extraction (ACE) program [15] as the task of finding relevant semantic relations between pairs of entities. Table 1 shows part of a document from ACE 2004 corpus (i.e. a collection of news articles). The text expresses the relation between the entity person, i.e. the *president* and the entity organization, *NBC's entertainment division*, where the person holds a managerial position.

Previous work has been devoted to automatically extract relations according to the ACE data and definitions. These models make use of machine learning and similarity measures over different features, which often take the form of kernel functions [35]. Such work has shown interesting and promising extraction systems, e.g. [10,13,38,40,41,44]. However, there are important aspects that need to be further developed and studied: first of all, the usual target language is English and there is a lack of indications if the previous approaches are cross-language or some modifications of them must be applied to deal with different language phenomena, such as richer morphology or freer argument syntax.

Secondly, the previous work on ACE provides results on standard texts, i.e. news items, whose complexity cannot be

Table 1 A document from ACE 2004 with all entity mentions in bold

Jeff Zucker, the longtime executive producer of NBC's "Today" program, will be named Friday as the new **president** of **NBC's entertainment division**, replacing Garth Ancier, NBC executives said

compared with more difficult data, e.g. non-standard free text, web pages, blogs and so on.

Thirdly, the relations between entities of the ACE program do not contribute to define any global information or aspect of its target domain, e.g. to determine/understand how the domain and its relationships are structured. For example, although there can be a bunch of relations related to Jeff Zucker (see Table 1), in the labelled ACE documents, we cannot hope to find all the relations between the employees of NBC, or the relations between the main actors of such TV broadcast domain, e.g. managers, anchor men, an so on.

Moreover, the use of academic benchmarks indirectly prevents to study models capable to deal with real application scenarios, e.g. noisy documents, whose content has been altered by unpredictable external conditions.

Finally, work on ACE only focuses on the extraction of the target relations and assumes that the set of named entities (NEs) participating to relations is already given in the targeted texts. This is not realistic since an effective system should automatically recognize, in general, the NEs from which the target relations can be extracted. The step of recognizing NEs, although can be automatically carried out with reasonably high accuracy, increases the complexity of RE.

In this paper, we carry out a study on the above-mentioned issues focusing on relation extraction (RE) from a real-world application perspective. In particular, we focused on complex relations between entities in textual documents from the investigative domain, where:

- the language of the application domain is Italian. To our knowledge, this is the first study on RE from text written in a language characterized by a rather rich morphology and argument free syntax.
- The textual data are constituted by reports on investigation into criminal organizations based on police interrogatory, electronic eavesdropping and wiretaps. The relations are typically defined among subjects mentioned in these sources, e.g. *person x belongs to criminal enterprise y* or *person x knows person y*. Entities and relations are rather specific and concentrated in the domain, and thus, they can define an overall content structure of the domain.
- The available text documents are manual transcriptions of dialectal utterances, where the inherently disfluencies usually affecting dialogs are mixed with the presence of non-standard lexicon and syntactic constructions. This data can be modeled as standard natural language text affected by noise, i.e. dialect expressions and transcription errors.
- Our automatic extractor is supposed to be used to speed-up and improve current investigation into real crime organizations. Thus, we closely consider its applicability in the target real scenario by also enabling relation extraction, when no labeled data, e.g. NEs, are available.

Our approach follows the typical machine learning setting for relation extraction: we used state-of-the-art statistical algorithms such as support vector machines along with effective and versatile pattern mining methods, e.g. word sequence kernels. However, the above-mentioned new characteristics of our application domain do negatively impact the accuracy of standard models forcing us to study and apply suitable solutions:

- first of all, the large presence of noise in the language used in our data prevented us to apply any syntactic parser. In Sect. 2.3, we show and discuss why any parser would provide unreliable outcome. This means that we needed to apply shallow syntactic and bag-of-words representations.
- Since previous approaches cannot deal with some instantiations of the studied textual relations, e.g. when entities are very distant (in terms of number of words) in the document, we engineered some specific features to help managing these difficult conditions, e.g. distance features.
- Most noticeably, to reduce the amount of training data required to obtain effective classifiers, we integrated domain prior knowledge from ontological data, e.g. type of relations and entity categories, in the form of logic constraints. Note that a similar articulated structure is not present in the specific ACE domains.

It should be also noted that our application scenario (and most likely many real-world scenarios) already includes an ontology of the relations expressed in the target domain. This ontology is simply generated by the analysts, who try to understand the content structure of the domain. Although the domain relation types are available, the work of an automatic relation extractor is still extremely important to derive relational information contained in new documents. In our case, the ontology is expressed by the relational schema of the database employed by analysts to store relations manually extracted by them while working in investigation. Such manual work also provided us with the needed training and test data.

We carried out several experiments to assess our models in the conditions of different levels of noise and different amount of training data. The results show that it is possible to build a relation miner, which is robust to non-conventional languages as dialects, jargon expressions or

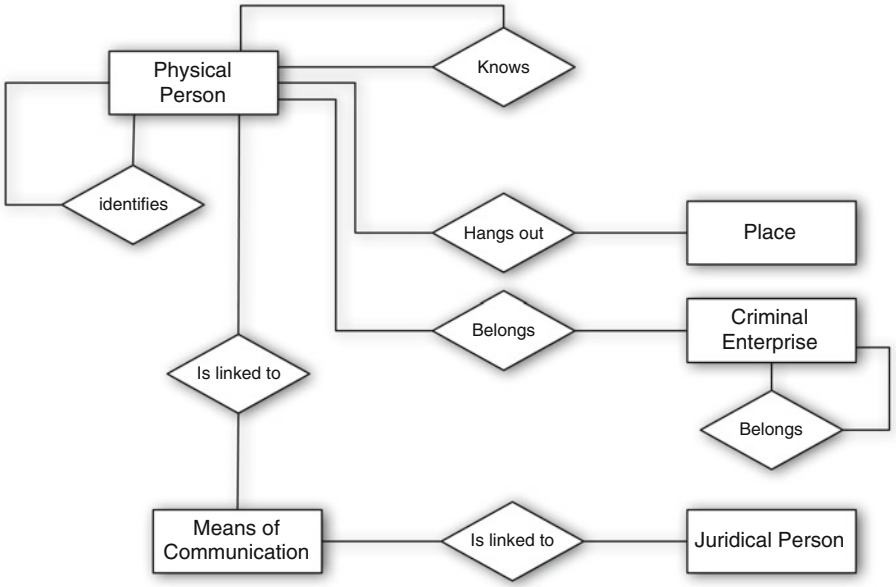

Fig. 1 The domain conceptual schema of relations treated in this research

coded words typically contained in our target intelligence text. Moreover, the experiments using different bins of training data show that our approach, based on ontological constraints, can achieve high accuracy even over small training data sets. In the remainder of this paper, Sect. 2 describes our target RE task along with the available corpus, whereas Sect. 3 introduces our RE system based on SVMs, kernel methods, innovative domain, specific features and ontological constraints. Section 4 illustrates our system evaluation along different dimensions: kernels, features, training data size and NE recognition. Finally Sect. 5 derives the conclusive remarks.

2 Relation extraction from investigative text

The research we present has been conducted in conjunction with the Italian Public Prosecutor's Office with the aim of designing automatic tools supporting countering organized crime. The investigation activities consist in analyzing huge amount of documents every day. These come from the investigative districts located in the whole national territory and report all the actions performed during investigation (e.g. arrests, trials, questioning reports and so on). A pool of experts analyze each document for finding evidence of relations among entities (e.g. *person* knows a *person* or *person* owns a *car*) with the intent to build connection nets among subjects, useful for intelligence activities. Such experts were also employed to build a computational corpus from which automatic REs can be learned and tested.

In addition to interesting characteristics of real scenario such as the relationship structure, which describes document content and provide information useful for carrying out investigative processes, there is the presence of several noise types. These range from utterance transcriptions to the use of jargon and disfluencies, which impact the lexicon and syntax of the target documents in a way similar to *more traditional* types of noise, e.g. mistakes in OCR processes. In the next section, we describe our task, corpus and the source of complexities also coming from the above-mentioned noise.

2.1 Domain relationships

Our relation extraction (RE) framework is formally identical to the one proposed in ACE. The task can be 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 text fragment observable in a document. The recognition of a given binary 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:

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

where the special symbol $\perp$ means no relation holds. In our definition, we stress the types of NEs, since in our domain they play an important role in the assignment of possible relations than in ACE. The analysts, which are domain experts, already know the relation types that are described by means of a conceptual schema, representing concepts and relationships of investigative interest.

The experts' task is to populate a DB during the analysis step. Figure 1 shows a subpart of the employed entity-relationships schema (used in our research). The relations described by such schema are briefly depicted in Table 2. For example, r_5 provides some interesting information for

Table 2 Relationship set from the relational DB in Fig. 1

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

Fig. 2 Excerpt of investigative document (questioning) containing relationship quotations. Named entities are highlighted in *bold*

.. Proprio mentre imboccavo la strada d'uscita vidi venire in senso contrario una Citroen scura con alla guida un tale **Mario** , uomo d'onore della famiglia mafiosa **Verdi** alla quale appartengono i **Rossi** di Milano, quelli proprietari di una pompa di benzina a Milano. A bordo vi era anche **Bianchi Giuseppe** ....

(a)

.... Posso affermare con certezza che alla cosca dei **Verdi** appartengono: **Mario Rossi**, **Giuseppe Verdi**, **Antonio Bianchi**, **Andrea Gialli** e i di lui fratelli **Nicola** e **Carlo**....

(b)

understanding the role of the different criminal organizations like CORLEONESI INCLUDES CALDERONE.

Database instances (i.e. the relation mentions) are populated with the data extracted from text. For example, the text fragment in Fig. 2 (from questioning transcription reports) includes two statements:[1]

(a) *Just when I was entering the way out street, I saw a darken Citroen driven by **Mario** arriving, a man belonging to the **Verdi**'s criminal enterprise to which the **Rossi** family from Milan belongs, those people owning a gas pump in Milan. On board there was also **Bianchi Giuseppe**…*

(b) *I can certainly claim that the **Verdi** family includes: **Mario Rossi**,[2] **Giuseppe Verdi**, **Antonio Bianchi**, **Andrea Gialli** and his brothers **Nicola** and **Carlo**…*

From the first statement, the relation r_4: MARIO BELONGS TO VERDI can be extracted. Moreover, another entity, *Bianchi Giuseppe*, is explicitly mentioned in the excerpt. From it, we understand that the two people above are traveling together in a car. Consequently, we can extract the relation r_1: MARIO KNOWS BIANCHI GIUSEPPE. It is worth noticing that this last relationship requires the interpretation of two subsequent sentences.

Another complex case is shown by the sentence (b) in which a list describes people belonging to a criminal enterprise. This relational structure can remarkably enlarge the distance between two related entities (we have observed enterprise's name and person at a distance up to 100 tokens). This is a critical difference with respect to other typical application domains of relation extraction. In the investigative domain, simplifying assumptions typical of standard RE as defined in ACE [18] are no longer valid.

The more general form on which relation can be realized is just one aspect of the complexity of our RE task. In the next sections, we describe all the other sources of complexity in detail.

2.2 Corpus construction

The design of RE corpus is a product of our collaboration with the Italian Public Prosecutor's Office. The same pool of experts, who manually analyzes documents for finding evidence of relations among entities, were also employed to carry out in line annotation, following guidelines similar to those adopted in the ACE program.

Our referring domain corpus is entirely written in Italian, and it is composed by rather heterogeneous types of documents as illustrated by Table 3. There are four different

[1] The translation reports the meaning of the original sentences. There is no attempt to show the disfluencies and ungrammaticality they contain. This will be discussed in the next section.

[2] This text fragment has been anonymized by using very common Italian names.

Table 3 Investigative corpus distribution

Document type	# of docs	Avg doc length (# of tokens)	Max doc length (# of tokens)	# of tokens	# of unique tokens
Informative reports	25	3,359.25	15,802	83,784	5,942
Printout transcriptions	8	3,295.42	9,817	27,020	4,213
Direct questioning reports	51	3,518.26	71,815	173,261	6,946
Summary of questioning	12	2,917.82	5,6317	35,002	4,634
Overall	96	3,518.26	71,815	319,067	8,937

types: Informative Reports, Printout Transcriptions, Direct Questioning Report and Summary of Questioning, where Questioning Reports, i.e. reports about a respondent person replying to specific questions or making spontaneous declarations, constitutes the majority of the documents.

The writing style is deeply different from the declarative prose largely characterizing ACE domains or from bioinformatics texts, which generally includes short and well-structured sentences. Our data consist of complex documents (e.g. printout transcriptions of mobile calls or the records of interrogatories), which do not follow any journalistic rule and are much more syntax free: the targeted relationship instances very often appear within an ill-formed sentence, which is full of syntactically illegal phenomena, or, even worse, they span across more than one such problematic sentences.

Moreover, the declarations made by involved actors can be rather vague or incomplete with also a large usage of jargon. The latter in the specific Italian case refers to dialectal expressions, which characterize declarations with a specific terminology and syntax. Given the large number of Italian dialects, a massive use of different linguistic varieties can be also found. This affects almost all document types, but in particular the printout transcriptions of mobile conversations.

Beside the complexity given by genuine linguistic phenomena, the targeted transcriptions also include large volumes of noise affecting the audio channel. Although we use manual transcription for our experiments, noise in the radio communication and dialect prevents human annotators to assess many fragments of conversations.

2.2.1 Corpus consistency

The team of analysts who has assisted us in this work was composed by six experts of domain (policemen and detectives). The process of corpus annotation was similar to classical annotation made by analysts during their work: for each instance of relationships found within text, they marked the involved entities and the text span in which the relationship is lexically realized. Although trained through very specific guidelines, annotators often do not follow them strictly. This is largely due to the combinatorial explosion of some phenomena, which are difficult to fully consider. Consequently, some cases are neglected, 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.* [3]

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 $6 \times (6-1)/2 = 15$ instances of the KNOWS relation) and between people and the location (i.e. Rome, with 6 instances of HANG OUT relation 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 was carried out. The analyst team was split in two different groups, each one annotated all test set documents. This allows us to compare the annotation[4] by means of various metrics, e.g. the inter-annotator agreement.

As discussed in Sect. 2.3, some complex problems affect 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, we evaluated inter-annotator agreement indices. Seven test documents were annotated by a second team (with analysts not included in the first team), which was trained according to the same modalities of the first team. After a short training on separate documents, they replicated the annotation so that all test cases were doubly annotated. The measure of the inter-annotator agreement observable between the two teams was the *Cohen's Kappa* [11], 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

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

[4] We could only carry out this more expensive annotation procedure on the test data. This is also convenient as we can accurately evaluate our systems although it has been trained with lower-quality annotations.

Table 4 Confusion matrix for relation r_1

	Team 1	
	Accepted	Rejected
Team 2		
Accepted	56	274
Rejected	0	10,509

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.[5] The inter-annotator agreement measures are reported in Table 5 according to κ values for each individual relation; high κ values are obtained for almost all relations.

The results show that annotation of relation r_1 (i.e. KNOWS) is much more controversial between the two teams. This is due to its combinatorial nature, which 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). This confirms the complexity of the targeted relation extraction task even for expert analysts.

A posterior analysis of the inter-annotator agreement scores suggested us that most of the disagreement was due to missed relations. Thus, in our final test set, we consider relations *if* almost a team has accepted it. Note that in this domain multiple relations between entities hold.

2.3 Complexity and noise in investigative data

The complexity of the document types is unfortunately fully reflected into the textual realizations of the targeted relationships. In the following sections, we outline three kinds of complexities: (1) structural, which concerns the content structure of the domain, (2) noise, which regards the quality of the sources of information and (3) the impact of the latter on the linguistic consistency, e.g. at syntactic and semantic level.

2.3.1 Structural complexity

The relationships useful for investigation and analysis are typically more complex than those defined in benchmarks. A prosecutor is interested to know whether there is a link between two people under investigation, even if this relation is broken down in two text fragments appearing in distant document points. Thus, a relation quotation usually includes a large number of tokens between the two mentions of the entities involved in the underlying relationship. Table 6 reports the statistics about the number of such tokens for individual relationships, which can be compared with the corresponding figures in the ACE 2004 training set (last row). Every relationship class of the investigative domain has a larger token distance, ranging from 16 to 32 tokens on average. This reflects the fact that relationships usually span over more than one sentence. Note that relationship r_2 (PP IDENTIFIES PP) is the only exception with an average token distance of 5. This happens as r_2 describes the situation in which a person *A* identifies another person *B* through the visual inspection of one of B's pictures, as provided by the detectives (this information is usually expressed with very short sentences). In contrast, the ACE relationships are realized in much shorter sentence fragments, typically with just one sentence. Indeed, the average distance between related entities is about 13 tokens: in particular, 44% entity pairs are not farer than 10 tokens, i.e. they belong to very short fragments.

Another complexity dimension relates to interpretation. This is even more open to subjectivity than standard natural language text. For example, a sentence like

> *Ne parlai con Mario e Giorgio* (I spoke to Mario and Giorgio about it).

was treated differently by individual annotators. One detects three instances of the relation KNOWS between the speaker, `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. In contrast, 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 each PP.

2.3.2 Noise types

The kind of noise affecting the investigative domain is different from the traditional noise present in other domains, e.g. speech data, OCR or user-generated contents [25,36]. The main difference is that in our case the noise is produced by human beings. Consequently, its analysis tends to be more complex in terms of the semantics of the noisy text fragments, e.g. sometime odd words can be considered noise produced by wrong transcriptions whereas in other cases the odd lexicals are just out-of-standard vocabulary words. In general, they may also assume a valid meaning by considering a specific pragmatic level.

[5] As discussed in [14], 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 by both the *bias* and *prevalence* problems. While we cannot avoid the bias problem, prevalence must be taken into account for interpreting the test outcome.

Table 5 Inter-annotator agreement according to *Cohen's Kappa*: Y*i* and N*i* refer to yes or not acceptance by the team *i*

	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 and Y2	56	9	53	50	3	6	10	187
Y1 and N2	0	0	0	0	0	1	0	1
N1 and Y2	274	1	27	4	1	0	0	307
N1 and 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 6 Distribution of the number of tokens between entities in a relationship quotation

id	Relationship	Min	Avg (±SD)	Max
r_1	PP KNOWS PP	2	23 (±9)	226
r_2	PP IDENTIFIES PP	3	5 (±3)	32
r_3	PP HANGS OUT PL	3	27 (±9)	246
r_4	PP BELONGS TO CE	3	32 (±15)	324
r_5	CE INCLUDES CE	1	22 (±0)	185
r_6	MC IS LINKED TO JP	2	16 (±6)	24
r_7	MC IS LINKED TO PP	1	19 (±3)	34
–	ACE_2004 training dataset	1	13 (±1)	54

Noise in our work reflects two major categories: *Recognition or Transmission Errors* due to the automatic processing of human-generated linguistic contents and *Uncertainty in the Communication process*, largely due to non-traditional and heterogeneous communication forms, e.g. similar to those in blogs or SMS-based communication. In the investigative domain, *Recognition or Transmission Errors* are due to the involvement of online sources that can be recorded in an imperfect environment:

- Transcription from telephone conversations, as included in the *Printout transcriptions* document types, is usually affected by a significant percentage of signal lacks and errors. This is mainly due to the environment conditions and audio low quality characterizing the recording session. Notice that the applied human intervention can partially alleviate the noise of the source, but cannot fulfill all lacks in the recordings. Usually, incomplete fragments are modeled through special tokens (e.g. `<INCOMPL>`) that are inserted in the transcriptions (e.g. *I was on a* `<INCOMPL>` *traveling to Rome with* `<INCOMPL>` *and we met …*)
- A specific form of noise is introduced by human beings, when reports are typed on the fly during the questioning sessions. These mistakes cannot always be removed through postanalysis and are usually left in the transcriptions. According to document types affected by these phenomena, i.e. the *Direct Questioning reports* as well as the *Summaries of Questioning* (which is about 66% of the corpus), this noise has a non-negligible impact on the overall accuracy. Notice that also Transcriptions from Telephone are seemingly affected by the same problems, so that this percentage is even larger.

In line with other notion of noise, as those studied in the literature (e.g. [25,36]), the investigative texts analyzed by this study are also affected by large amounts of noise due to the *Uncertainty in the Communication process*. In particular:

- The conversational nature of the questioning sessions reflects in a rather informal reply of the respondents, either in *Direct Questioning* reports or in *Summaries of Questioning*. This material is characterized by a strong presence of dialectal forms and jargon that makes it close to phenomena typical of ungrammatical texts. In other words, it is not possible to devise a reference grammar for these heterogenous materials as they are originated from different sublanguages (such as different dialects for different geographical areas).
- The terminological variability of the targeted textual phenomena is also very high as the different dialects provide very different expressions (typical of different subcommunities) for the same phenomena. It is mandatory to adopt specific data-driven models just to integrate this information in the knowledge available to the relation recognition system.
- An important form of noise emerges when a relationship instance (i.e. the pair of its involved entities) spans over more than one sentence, i.e. a multisentence span. In this case, the amount of information irrelevant to the relationship increases linearly in the number of tokens. This corresponds to information irrelevant to the relation extraction task, which acts as a noise for the recognition process. The investigative domain, as reflected in Table 6,

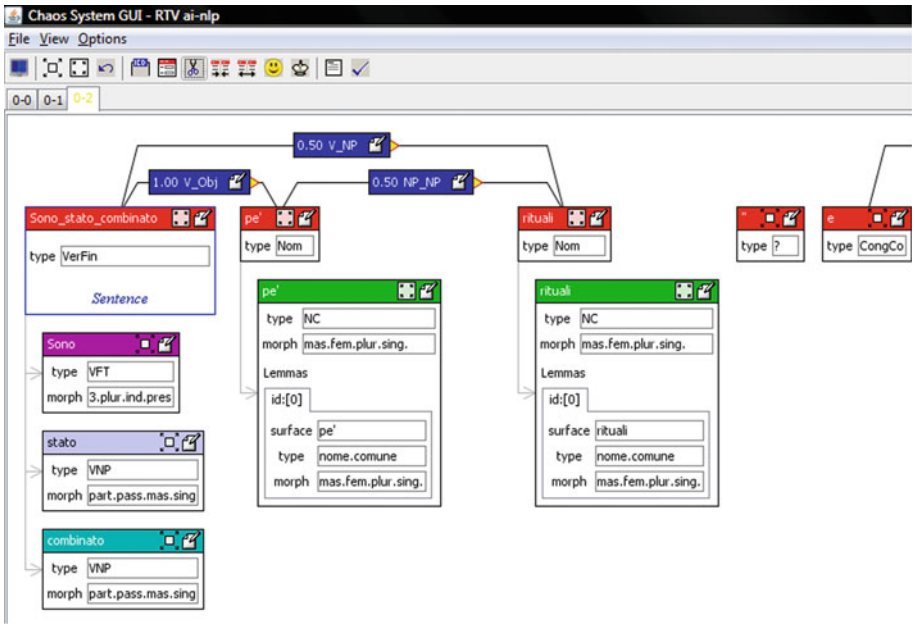

Fig. 3 The syntactic analysis of a short fragment of a typical direct questioning report for the sentence: *Sono stato combinato pe' rituali e il mio padrino e' stato il John Doe*

is affected by these phenomena in a much stronger fashion than previous existing relation extraction tasks (such as ACE).

2.4 Noise impact to linguistic complexity

The noise described in the previous section deeply impacts the linguistic processors that we can use to implement our RE system. We characterize such impact in the following analysis.

First of all, we note that the natural language phenomena occurring in our 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 suggests that the application of a syntactic parser is unhelpful as for coverage at the level of lexical and grammatical phenomena.

As an example, let us consider the following short sentence of our corpus:

> *Sono stato* ***combinato pe' rituali*** *e il mio padrino é stato il John Doe*
> (I have been introduced through the usual rituals and my godfather was John Doe)

The words in bold show dialectal lexicon phenomena that affect the syntax of the entire sentence and produce an incorrect syntactic interpretation. In Italian, the word *combinato* has no meaning related to the ceremony for initiating an individual to a group, expressed mostly by the verb *iniziare* (i.e. *to initiate*). Moreover, also the expression **pe' rituali** is odd; it is a jargon expression for the meaning *attraverso i (soliti) rituali* (through the (usual) rituals). Here *pe'* plays the role of preposition and should be interpreted as a prepositional modifier of the main verb (*combinato*).

Figure 3 reports the syntactic parse of the fragment above, automatically generated by our Italian parser called CHAOS [3]. The text fragment, *Sono stato combinato pe' rituali* ... in the morpho-syntactic boxes (i.e. 2nd level boxes), shows that the words *pe'* and *rituali* are interpreted as nouns ($POStag$ = NC). Consequently, the interpretation of their dependency with the main verb *combinato* (i.e. *initiated*) is wrong, i.e. *pe'* is interpreted as a direct object (V_obj).

In more detail, all three grammatical relations generated from the fragment *combinato pe' rituali* are wrong, as they fail to capture the prepositional attachment between *combinare* (i.e. *to initiate*) and *rituali* (i.e. *rituals*): no semantic interpretation is thus made available for correctly detecting the initiation event.

Moreover, given the entire syntactic graph, it is even difficult to establish a relationship between the speaker (i.e. the intended subject of the main verb (*I've been initiated*) and *John Doe*, as the graph is not fully connected. This makes the extraction of the semantic relationship r_1 (PP KNOWS PP) between the respondent (i.e. the implicit person of the Direct Questioning report) and *John Doe*, impossible.

3 A relation extraction system for investigative text analysis

Our mining system architecture follows the classical relation extraction (RE) models, e.g. [9,13,22,40,45]. We automatically learn the RE function f in Eq. 1 from data. This decides if two target entities e_i and e_j are in a target relationship.

For this purpose, the entity pair is mapped into a vector $\mathbf{x}_{ij}$ of properties expressing different types of features of the text unit t_{ij} (i.e. a potential quotation) in which they appear. A binary classifier for each relation r_k can be learned from existing repositories of annotated examples. We combine the set of binary classifiers in the multiclassifier f with the *one vs. all* method [32]. To map t_{ij} in vectors, we use manually designed features (linear kernels) as well as implicit mapping given by sequence kernels, e.g. [9].

In addition to the two above standard methods, we:

- carry out experiments with both gold standard and automatic NEs. Although this should be a straightforward step in applied RE, most previous work on RE from ACE corpora does not use automatic NEs (e.g. [9,13,43]) i.e. the NEs manually annotated are utilized for evaluating the final accuracy. This produces very different results from a completely automatic setting.
- encode prior knowledge in the classification system by means of the ontology constraints derived by the database schema in Fig. 1, which is designed to contain the target relations. In more detail, (1) we apply our named entity recognizer, which detects the target entity mentions; (2) then all possible pairs of entities are generated; (3) we impose the logic constrains coming from the entity categories and the DB schema to filter out invalid relationships; and (4) we apply the relation multiclassifier to the remaining pairs.

In the next sections, we described our models in more detail.

3.1 Kernels and support vector machines

Kernel methods (e.g. see [35]) refer to a large class of learning algorithms based on inner product vector spaces, among which support vector machines (SVMs) [37] are one of the most well-known algorithms. SVMs learn a hyperplane $H(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{x}$ is the feature vector representation of a classifying object o, $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are parameters [37]. The classifying object o is mapped into $\mathbf{x}$ by a feature function ϕ. The kernel trick allows us to rewrite the decision hyperplane as $\sum_{i=1,\ldots,l} y_i \alpha_i \phi(o_i)\phi(o) + b = 0$, where y_i is equal to 1 for positive and -1 for negative examples, $\alpha_i \in \mathbb{R}^+, o_i \forall i \in \{1,\ldots,l\}$ are the training instances and the product $K(o_i, o) = \langle \phi(o_i)\phi(o) \rangle$ is the kernel function associated with the mapping ϕ. Note that we do not need to apply the mapping ϕ, we can use $K(o_i, o)$ directly. This allows us, under the Mercer's conditions [35], to define abstract kernel functions that generate implicit feature spaces, where the SVM optimization algorithm is guaranteed to converge to a global optimum according to the geometric interpretation of margin maximization.

Moreover, 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 simple operations on kernels: $K_1 + K_2$ or $K_1 \times K_2$. These combinations are very useful to mix the knowledge provided by the original features, for example acting on different perspectives (e.g. lexical vs. syntagmatic) on the original objects, e.g. textual units.

In next section, we illustrate a sequence kernel function that counts the number of word sequences in common between two sentences, in the space of n-grams (for any n) by also considering gaps.

3.1.1 Word sequence kernels

The Word Sequence Kernels that we consider count the number of subsequences of words containing gaps shared by two sequences, i.e. some of the symbols of the original sequence are skipped. Gaps modify the weight associated with the target subsequences as shown in the following.

Let Σ be a finite alphabet, $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$ is the set of all word sequences. Given a string $s \in \Sigma^*$, $|s|$ denotes the length of the strings and s_i its compounding symbols, i.e $s = s_1, \ldots, s_{|s|}$, whereas $s[i : j]$ selects the substring $s_i s_{i+1}, \ldots, s_{j-1} s_j$ from the ith to the jth character. u is a subsequence of s if there is a sequence of indexes $\mathbf{I} = (i_1, \ldots, i_{|u|})$, with $1 \leq i_1 < \cdots < i_{|u|} \leq |s|$, such that $u = s_{i_1}, \ldots, s_{i_{|u|}}$ or $u = s[\mathbf{I}]$ for short. $d(\mathbf{I})$ is the distance between the first and last character of the subsequence u in s, i.e. $d(\mathbf{I}) = i_{|u|} - i_1 + 1$. Finally, given $s_1, s_2 \in \Sigma^*$, $s_1 s_2$ indicates their concatenation.

The set of all substrings of a text corpus forms a feature space denoted by $\mathcal{F} = \{u_1, u_2, \ldots\} \subset \Sigma^*$. To map a string s in $\mathbb{R}^\infty$ space, we can use the following functions: $\phi_u(s) = \sum_{\mathbf{I}:u=s[\mathbf{I}]} \lambda^{d(\mathbf{I})}$ for some $\lambda \leq 1$. These functions count the number of occurrences of u in the string s and assign them a weight $\lambda^{d(\mathbf{I})}$ proportional to their lengths. Hence, the inner product of the feature vectors for two strings s_1 and s_2 returns the sum of all common subsequences weighted according to their frequency of occurrences and lengths, i.e.

$$\mathrm{SK}(s_1, s_2) = \sum_{u \in \Sigma^*} \phi_u(s_1) \cdot \phi_u(s_2) = \sum_{u \in \Sigma^*} \sum_{\mathbf{I}_1 : u = s_1[\mathbf{I}_1]} \lambda^{d(\mathbf{I}_1)} \sum_{\mathbf{I}_2 : u = s_2[\mathbf{I}_2]} \lambda^{d(\mathbf{I}_2)} = \sum_{u \in \Sigma^*} \sum_{\mathbf{I}_1 : u = s_1[\mathbf{I}_1]} \sum_{\mathbf{I}_2 : u = s_2[\mathbf{I}_2]} \lambda^{d(\mathbf{I}_1) + d(\mathbf{I}_2)},$$

where $d(.)$ counts the number of characters in the substrings as well as the gaps that were skipped in the original string. It is worth noting that (a) longer subsequences receive lower weights; (b) some characters can be omitted, and gaps

determine a weight since the exponent of λ is the number of characters and gaps between the first and last character.

Characters in the sequences can be substituted with any set of symbols. In our study, we preferred to use words so that we can obtain word sequences. For example, given the sentence: *Mario Rossi is affiliated to the Corleone's family* sample substrings, extracted by the Sequence Kernel (SK), are: *Rossi is affiliated Corleone's*, *Rossi affiliated family*, *Rossi affiliated*, *Rossi Corleone's*, etc.

3.2 Specific representation for investigative data

Section 2 has shown that in our domain there are linguistic complexities and noise, which prevent the use of parser and of other standard kernels. We hereafter provide our relation representations, which make RE applicable to our data.

3.2.1 Sequence Kernels

Word Sequence 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. In particular, in our work, we use an approach similar to [9], who builds three different sequences for any quotation:

- a *Fore-Between* segment (FB) is made by the first n words before and the first n words after the first entity in the text;
- the *Between* segment (B) is made by the words appearing between the two entity mentions; and
- the *Between-After* segment (BA) includes the first n words before and the first n words after the second entity in the text;

These three sequences are used to feed three different sequence kernels, whose final contribution is summed together. Unfortunately, the relations of our domain tend to be realized between entities even at a very long distance in the text as we pointed out in Sect. 2.3.1. To cope with this higher complexity, we only use the two FB and BA sequences. Note that, by choosing an enough big n the two sequences include the three original sequences. In case of very long distances, e.g. entities spanned over multiple sentences, the resulting kernel only acts on text fragments that are local to e_i and e_j, i.e. our kernel tends to capture shallow (local) syntactic information. It will be hereafter referred as K_{Seq}.

It should be noted that (1) in our experiments the two-window-based kernels outperform the original three-window representations (see discussion in Sect. 4.4) and (2) the information between the two entities, which is lost in case of long distance NEs, is partially recovered by our manually designed features presented in the next section.

3.3 Manually designed features

Along with kernel methods, we manually design a set of effective attributes illustrated hereafter:

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

Entity Types. In order to increase the generalization power of individual features, entities (e.g. *Mario*, *Roma*) are substituted by their corresponding class (e.g. person). For example in the excerpt

> *Lui ha abitato a Roma per un periodo* (He has lived in Rome for a while)

the active tokens in the representation are {PP, *ha, abitato, a*, PL, *per, un, periodo* }, where PP and PL indicate physical person and place, respectively.

Distance between mentions to entities. We use the distance between the two entities as a filter for candidate pairs and we also define three different features: short, medium and large distance associated with the three main percentile, 33, 66 and 100% on the distributions of distance values of correct instances, respectively. Thus, each feature characterizes valid relations, with a decreasing precision.

Punctuation. Punctuation in the *Fore*, *Between* and *After* text portions is represented via special features. These account for the position of each punctuation mark with respect to the two entities. For example, a comma in the *Fore* text component (i.e., before the entity e_i) is denoted by `#,F`, while `#,B` is reserved for commas appearing between the two entity mentions. Moreover, each feature is weighted by its frequency.

Ordering of mentions. This is a boolean feature, Ord, which indicates if the entities appear in the text fragment with the same order indicated by the target relation r_k. For example, the HANGS OUT relation is clearly orientated from people PP to places PL, whereas the fragment *A Roma l'incontro con Mario si protrasse sino a tarda notte* (In Rome, the meeting with Mario lasted until late night) expresses the two entities in the reverse order with respect to r_k: in this case, the feature Ord assumes the `false` value.

3.4 Kernel combinations

Our baseline model is based on the linear kernel, K_{BOW}, applied to vectors built with only lexical units. When all the other manually designed features are included in the vector, we call the related model, K_{XBOW}, i.e. the extended kernel. We can combine the functions above with sequence kernels. This way, lexical and syntagmatic spaces are modeled independently, via K_{BOW} (or K_{XBOW}) and K_{Seq} respectively.

The overall kernel is defined through the usual sum:[6] $K(X_1, X_2) = K_{\text{XBOW}}(X_1, X_2) + K_{\text{Seq}}(X_1, X_2)$.

3.5 Prior knowledge via ontological constraints

Disfluencies and jargon expressions occurring in the target text highly increase the complexity of the textual model that statistical methods have to learn. Thus, reliable classifiers can be only achieved using large amount of training data. This is a major drawback in practical applications since costs and time constraints prevent to rely on large corpora. To reduce the amount of the required data, we exploit background knowledge under the form of relational schema[7] of the underlying database containing the relational instances.

In more detail, our ontological filters work as follows. Our annotated text corpus contains instances 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, r_k, is also a negative example for every relation r_l ($l \neq k$) that insists on the same entity pairs[8] 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 schema in Fig. 1 for a full description).

However, negative training examples also include pairs of entities that are not in a relationship. In order to build the full set of negative examples for a target relation, we computed all possible entity pairs from a document that: (1) are not positive examples of such relation and (2) appear in at least one relationship class in the domain schema (i.e., there is an entry in Table 3). This assumption states that every candidate quotation, which is a feasible candidate entity pair in the DB schema, is a negative example only when no annotation is available for it. Note that the above assumption limits the set of candidate pairs, although they may proliferate in long documents.

Additionally, to deal with the above-mentioned proliferation, we impose a thresholds on the maximal distance allowed between two entities e_i and e_j. The analysis of the annotated corpus showed that most of the entity pairs in valid relation instances generally occurred within a limited distance.[9] 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, estimated over the training set.[10] 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.

3.6 Related work

To identify semantic relations using machine learning, three learning settings have mainly been applied, namely supervised methods, e.g. [13,22,26,40,45], semi-supervised methods, e.g. [1,8], and unsupervised method, e.g. [21]. In a supervised learning setting, representative related work can be classified into generative models, e.g. [26], feature-based, e.g. [22,33,44,45] or kernel-based methods, e.g. [10,13,38,40–42].

The learning model employed in [26] used statistical parsing techniques to learn syntactic parse trees. It demonstrated that a lexicalized, probabilistic context-free parser with head rules can be effectively used for information extraction. Generally, feature-based approaches often employ various kinds of linguistic, syntactic or contextual information and integrate them into the feature space. Roth and tau Yih [33] applied a probabilistic approach to solve the problems of named entity and relation extraction with the incorporation of various features such as words, their part-of-speech, and semantic information from WordNet. Kambhatla [22] employed maximum entropy models with diverse features including words, entity and mention types and the number of words (if any) separating the two entities.

Recent work on Relation Extraction has mostly employed kernel-based approaches over syntactic parse trees. Kernels on parse trees were pioneered by Collins and Duffy [12]. This kernel function counts the number of common subtrees, weighted appropriately, as the measure of similarity between two parse trees. Culotta and Sorensen [13] extended this work to calculate kernels between augmented dependency trees. Zelenko et al. [40] proposed extracting relations by computing kernel functions between parse trees. Bunescu and Mooney [10] proposed a shortest path dependency kernel by stipulating that the information to model a relationship between two entities can be captured by the shortest path between them in the dependency graph.

Although approaches in RE have been dominated by kernel-based methods, until now, most of the research in this line has used the kernel as some similarity measures over diverse features [10,13,38,40,41]. A recent approach successfully employs a convolution tree kernel over constituent syntactic parse tree [42,46]. The combination of such kernel

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

[7] The availability of such schema (or other ontological schema) is not a strong assumption since a database is typically used to store intelligence data. The problem may arise when designing a new application from scratch.

[8] This is true in our domain since the relations are mutually exclusive.

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

[10] We choose the 90th percentile since it maximizes coverage while minimizing the number of false instances introduced.

with others based on grammatical relations from dependency structure was successfully modeled in [30].

As shown in the Sect. 2.3, we cannot apply any kind of parsing in our target domain; this prevents the use of the findings in most part of the mentioned researchers since they are deeply based on syntactic trees.

4 Experiments

We evaluated the impact of our relational miner on a real test collection. Our objectives were: (a) to provide a comparative analysis of different learning algorithms and representation models by measuring the reachable accuracy; (b) to study the properties of robustness to noise and lack of training data of our approach; (c) to measure the impact of fully automatic RE based on named entity recognition; and (d) to measure the impact of our ontological constraints.

4.1 Setup

The experimental corpus was derived from two collections of public judicial acts related to the legal proceedings against the same large criminal enterprise. It is constituted by 96 documents, annotated by analysts. We used 82% (i.e., 79 documents) for training and 8% (7 documents) for testing. The remaining 10% (10 documents) has been used as a development set to optimize the parameter settings for all the compared algorithms. Note that we double the annotation on test documents to produce a more accurate test set. This prevents us to carry out cross-validation as the quality of the training data is much lower and cannot be used to reliably measure accuracy.[11]

Although the manual annotation was carried out on 15 different relationship classes, some of them were rather rare; this prevented us to use them. Thus, the experimentation was only focused on the seven relations reported in Table 3. Skewed distributions can be observed, where some relations are much more common in documents like PP HANGS OUT AT A PL or PP KNOWS PP and others 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.

The experimental corpus is described in Table 7. 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 were 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 × PP)) coincides.

Table 7 Experimental data set

Id Id	Relationship class	Training instances (% of positives)	Test instances
r_1	PP KNOWS PP	3,985 (16.18)	519
r_2	PP IDENTIFIES PP	3,985 (5)	519
r_3	PP HANGS OUT PL	2,359 (14.83)	229
r_4	PP BELONGS TO CE	1,717 (35.11)	103
r_5	CE INCLUDES CE	604 (20.19)	10
r_6	MC IS LINKED TO JP	62 (51.6)	22
r_7	MC IS LINKED TO PP	231 (42.85)	39

A final very important remark regards the application of ontological constraints as explained in Sect. 3.5. We always apply such constraints in all our experiments. We attempted to disable such feature but we obtained very low Micro-average F1, i.e. about 40%. This confirms that the use of background knowledge is extremely important in case of scarce training data availability.

4.2 Comparative analysis

In this section, we present comparative experiments to analyze the impact of different feature models across a set of learning algorithms.

We applied two well-known learning algorithms, i.e. C4.5 decision trees [31] and Naive Bayes to our data sets.[12] Both systems were run over the feature set characterizing the K_{XBOW} kernel (i.e., bag-of-words extended with the domain features discussed in Sect. 3.3). Additionally, we provide a baseline accuracy given by random choices across the candidate pairs (filtered according to the 90th percentile statistics).

All the algorithms were optimized over the same development set and then tested against the data shown in Table 7. For evaluation, we used the classical evaluation metrics: Precision (i.e. the percentage of correctly recognized relation instances against the total number of accepted test cases), Recall (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 was also measured as the percentage of correct recognition inferences, thus including the acceptance of correct candidates and the rejection of false candidates.

The comparative evaluation of different algorithms trained with the best parameterization (over a held-out set) is shown

[11] The fact that we successfully use these poorly annotated training data to learn our model is another interesting finding of our research.

[12] Both algorithms have been tested through Weka [39].

Table 8 Comparative evaluation (micro-average) among classification algorithms

Algorithm	Precision	Recall	F1	Acc (%)
Random choice	0.13	0.4	0.19	41
Decision tree	0.45	0.24	0.31	54
NaiveBayes	0.34	0.56	0.42	57
K_{BOW}	0.32	0.75	0.45	66
K_{XBOW}	0.70	0.83	0.75	85
$K_{\text{XBOW}} + K_{\text{Seq}}$	0.75	0.85	0.80	88

Table 9 F-measure score of the SVM models over individual relationship classes

Id	Relationship class	K_{XBOW}	$K_{\text{XBOW}} + K_{\text{Seq}}$
r_1	PP KNOWS PP	0.40	0.52
r_2	PP IDENTIFIES PP	1.00	1.00
r_3	PP HANGS OUT at a PL	0.40	0.68
r_4	PP BELONGS to CE	0.66	0.75
r_5	CE INCLUDES CE	1.00	1.00
r_6	MC IS LINKED TO JP	0.70	0.70
r_7	MC IS LINKED TO PP	1.00	1.00

in Table 8. The last three rows represent the systems trained with different kernels.[13]

The results show that the Precision of Decision Tree and NaiveBayes is better than the one of K_{BOW} (i.e. the SVM simple model) but their F1s, i.e. 0.31 and 0.42, are lower than the F1 of K_{BOW}, i.e. 0.45. This is due to the higher generalization power of SVMs. The best models are K_{XBOW} and $K_{\text{XBOW}} + K_{\text{Seq}}$, which can exploit our extended features and sequence kernels, reaching the interesting values of 0.75 and 0.80, respectively.

4.3 Feature analysis

The good results obtained through the different kernels, as shown by Table 8, inspired an analysis of the impact of the different models over the individual relations. As discussed in Sect. 3.3, 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.

Note 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, we ran several tests. The best trade-off between Precision and Recall scores was achieved with the following feature configuration:

- *Lexical Units*: tokens.
- *Entity Types*: textual mentions to entities (e.g. $Mario$) are substituted with their corresponding type labels (e.g. 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 9, the F-measure scores as obtained for individual relations according to the above XBOW model are reported. Most of the relations obtain an ex cellent 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 performance, basically due to the presence of dialectal or syntactically odd expressions. The combination of the two kernels (last column of Table 9) seems to overcome most of these problems.

Notice that the weakest 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 $K_{\text{XBOW}} + K_{\text{Seq}}$ 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 higher value of 0.82 (vs. 52 %).

As a final test, we computed the Precision/Recall curve for $K_{\text{XBOW}} + K_{\text{Seq}}$ model, reported in Fig. 4. The Precision/Recall curves were built varying the learning parameter J of SVM-light-TK, i.e. the relative weight to positive instances with respect to negative instances. The plot shows a regular shape and 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.

4.4 Analysis of learning ability and robustness to noisy

As previously discussed in Sec. 3.2.1, our models provide an important contribution in case of noisy and complex data

[13] For the SVM learning, we used the SVMlightTK platform as available at: http://dit.unitn.it/~moschitt/Tree-Kernel.htm.

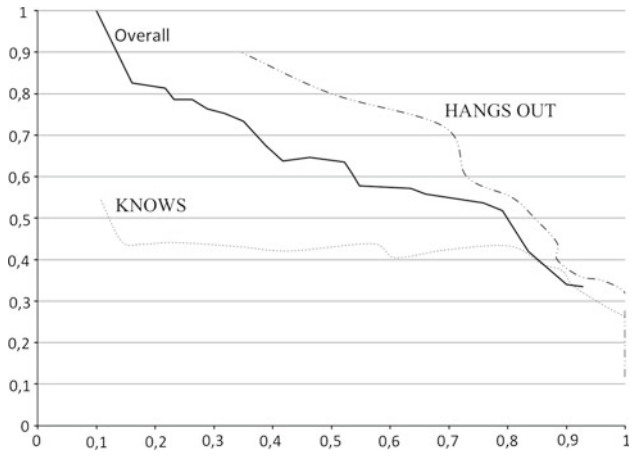

Fig. 4 Precision/recall *curve* for the relationship classes KNOWS and HANGS OUT and for the global relation extraction system

which cannot be managed with the classical data representation used for academic benchmarks. In fact, the use of only two token windows instead of three allows to learn over examples not otherwise computable. In order to prove the robustness of our model, two further kernel representations have been tested.

These representations are based on the standard representation of quotations through three text windows as defined in [9]. In these experiments, the problem of long distance among entities is overcome by substituting the *Between* window with a special window when the former is larger than a fixed threshold. Two special windows have been adopted:

- Break window: when the distance between entities exceeds a certain threshold, the text segment between the two entities is substituted by the token *#Break*. Thus, only the external (to the NEs) contexts of the entities will be considered.
- Random window: the *Between* segment is formed by a fixed number of tokens that are randomly selected from the original sequence. Each token is followed by a special token *#Random* that defines this representation.

As shown in Table 10, the combinations of the above kernels with XBOW do not improve the results obtained with our word sequence kernel (applied to the two text windows centered in the target NEs); in other words, these representations do not provide the same robustness on complex data.

To study the benefit of our hybrid model on training data requirement, we report the learning curves of the seven relation classifiers in Fig. 5. It is interesting to note that with 50% of training data (corresponding to 40 documents) all the classifiers almost reach a plateau, with an F1 ranging from 50% to 100%. The most interesting aspect is that the maximum accuracy, which is also the state-of-the-art for such complex mining task, can be reached with relatively few training documents.

Table 10 Comparative evaluation (micro-average) among different kernel representations

Kernel	Precision	Recall	F1	Acc (%)
BOW	0.32	0.75	0.45	66
K_{XBOW}	0.70	0.83	0.75	85
$K_{\text{XBOW}} + K_{\text{BMBreak}}$	0.56	0.83	0.66	70
$K_{\text{XBOW}} + K_{\text{BMRandom}}$	0.69	0.83	0.75	76
$K_{\text{XBOW}} + K_{\text{Seq}}$	0.75	0.85	0.80	88
NER + $K_{\text{XBOW}} + K_{\text{Seq}}$	0.77	0.682	0.72	74

"NER+" label indicates that the NEs are derived by means of automatic named entity recognizer

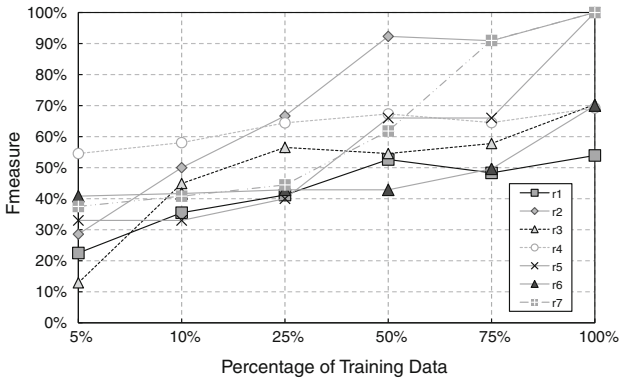

Fig. 5 Learning *curves* for the target relations

4.5 End-to-end system evaluation

Another important evaluation regards the fully automatic relation extraction, i.e. where also the named entities are automatically derived. We used our re-implementation of the well-known NER Identifinder [4]. This is based on a simple hidden Markov model with smoothing and multiple backoffs. The adopted features are accurately described in [4]. They mainly characterize the strings constituting the language model (e.g the string is in upper case, the initial letter of the string is capitalized, the string is alphanumeric and so on). The training instances for our NER are generated from the same data used for training the RE systems: each relation indeed has necessarily annotated its arguments, i.e. the two NEs.

The last row of Table 10, i.e. NER + $K_{\text{XBOW}} + K_{\text{Seq}}$, shows the F1 (0.72) of our best RE model, when NEs are automatically extracted. This datum can be compared with the result using gold standard NEs, i.e. 0.80. Table 11 shows the F1 of the classifiers for the target seven relations. The results demonstrate that our approach is robust to the noise produced by the NER since the F1 of the different relations are very near to those achieved with gold standard NEs (see Table 9). This is also due to the *good* accuracy of our NER on the target domain, as shown in Table 12.

Table 11 F1-measure over individual relationship classes, when named entity are automatically derived

	Precision	Recall	F1
r_1	0.44	0.62	0.52
r_2	1.00	1.00	1.00
r_3	0.53	0.59	0.56
r_4	0.43	0.33	0.38
r_5	1.00	0.70	0.77
r_6	1.00	0.54	0.70
r_7	1.00	1.00	1.00

Table 12 F1-measure of named entity recognizer

	Precision	Recall	F1
Physical person	0.79	0.81	0.80
Criminal enterprise	0.62	0.82	0.70
Place	0.72	0.92	0.81
Juridical person	0.91	0.77	0.83
Means of communication	0.88	0.94	0.91

5 Conclusive remarks

One interesting data mining problem is relation extraction between entities in textual documents. We have presented robust models for linguistic relation mining from a business intelligence domain, where reports on criminal investigation, police interrogatory, electronic eavesdropping and wiretap constituted the typical data. The relations to be mined occur between subjects mentioned in documents. This application scenario is highly affected by linguistic noise and by the complexity of natural language data.

Our solution is based on (1) supervised approaches, i.e. support vector machines along with effective and versatile pattern mining methods, e.g. sequence kernels; (2) design of new specific features to deal with the generality of the target application domain; and (3) the exploitation of the ontological information extracted by the relational schema of the underlying database used by the manual investigative approach.

Our collaboration with the investigative team allowed us to leverage the previous manual work to derive the ontology and the annotated data[14] that we used to design and test our models.

To measure the impact of our models, we carried out several experiments: (1) for measuring the complexity of our produced corpus by means of the inter-annotator agreement score; (2) to compare different models using different kernels; (3) to study the robustness with respect to data availability and noise.

The results show that:

- The sequence kernel along with the manually designed features provides the highest accuracy which is rather satisfactory, i.e. an F1 of 79%.
- The learning curves show that the best model needs only 40 training documents to reach a plateau, suggesting that a fast design using small training data is viable.
- The fully automatic task, which includes the use of a named entity recognizer, demonstrates the robustness to the noise of our approach since the system achieves an F1 of 72%, corresponding to a relatively small accuracy decay.
- The ontological constraint proved to be essential as when we disable them we obtain very low accuracy (an F1 of about 40%).

It should be noted that our approach is state-of-the-art for this task since the best models in relation extraction, e.g. [9,13,21,22,43], are not applicable in our case. Indeed, disfluencies (make dependency and constituent parsing difficult to apply) and types of relations (e.g. spanning different paragraphs) require a completely different design of the above-mentioned approaches in order to be applied to our data.

However, we do believe that the design of robust approaches able to exploit deeper syntax and shallow semantics is an interesting research line. Advanced representations based on predicate argument structures, e.g. [19,20,28,29], may result robust to noise and provide the required syntactic and shallow semantic information as it has been already shown in [27]. Additionally, term similarity kernels, e.g. [2,5], will be likely to improve relation generalization, especially when combined syntactic and semantic kernels are used, i.e. [6,7].

[14] We are planning to make it available, where the NEs and other sensible information is ciphered.

References

1. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the 5th ACM International Conference on Digital Libraries (2000)
2. Basili, R., Cammisa, M., Moschitti, A.: Effective use of WordNet semantics via kernel-based learning. In: Proceedings of CoNLL-2005, Ann Arbor, Michigan (2005)
3. Basili, R., Zanzotto, F.M.: Parsing engineering and empirical robusteness. J. Lang. Eng. **8**(2/3) (2002)
4. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An algorithm that learns what's in a name. Mach. Learn. **34**(1–3), 211–231 (1999)
5. Bloehdorn, S., Basili, R., Cammisa, M., Moschitti, A.: Semantic kernels for text classification based on topological measures of feature similarity. In: Proceedings of ICDM'06, Hong Kong, 2006 (2006)

6. Bloehdorn, S., Moschitti, A.: Combined syntactic and semantic kernels for text classification. In: Proceedings of ECIR 2007, Rome, Italy (2007)
7. Bloehdorn, S., Moschitti, A.: Structure and semantics for expressive text kernels. In: In Proceedings of CIKM '07 (2007)
8. Brin, S.: Extracting patterns and relations from world wide web. In: Proceeding of WebDB workshop at 6th international conference on extending database technology, pp. 172–183 (1998)
9. Bunescu, R.C., Mooney, R.J.: Subsequence kernels for relation extraction. In: Proceedings of NIPS (2005)
10. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of EMNLP, pp. 724–731. (2005)
11. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Comput. Linguist. **22**(2), 249–254 (1996)
12. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS'2001) (2001)
13. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on ACL, Barcelona, Spain (2004)
14. Di Eugenio, B., Glass, M.: The kappa statistic: a second look. Comput. Linguist. **30**(1), 95–101 (2004)
15. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: The automatic content extraction (ACE) program—tasks, data, and evaluation. In: Proceedings of LREC 2004, pp. 837–840. (2004)
16. Dzeroski, S., Blockeel, H.: Introduction to the workshop. In: MRDM'05: Proceedings of the 4th International Workshop on Multi-Relational Mining, New York, NY, USA, ACM (2005)
17. Dzeroski, S., Lavrac, N. (eds.): Relational Data Mining. Springer, New York (2001)
18. Gabbay, I., Sutcliffe, R.F.: A qualitative comparison of scientific and journalistic texts from the perspective of extracting definitions. In: Aliod, D.M., Vicedo, J.L. (eds.) ACL 2004: Question Answering in Restricted Domains, pp. 16–22. Association for Computational Linguistics, Barcelona, Spain, July 2004
19. Giuglea, A.-M., Moschitti, A.: Knowledge discovery using framenet, verbnet and propbank. In: Meyers, A. (eds.) Workshop on Ontology and Knowledge Discovering at ECML 2004, Pisa, Italy (2004)
20. Giuglea, A.-M., Moschitti, A.: Semantic role labeling via framenet, verbnet and propbank. In: Proceedings of ACL 2006, Sydney, Australia, 2006
21. Hasegawa, T., Sekine, S., Grishman, R.: Discovering relations among named entities from large corpora. In: Proceedings of the 42nd Annual Meeting on ACL, Barcelona, Spain (2004)
22. Kambhatla, N.: Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on Interactive poster and demonstration Sessions, Barcelona, Spain (2004)
23. Kim, J.D., Ohta, T., Tateisi, Y., Tsujii, J.: Genia corpus—semantically annotated corpus for bio-textmining. Bioinformatics **19**(1) (2003)
24. Kolak, O., Schilit, B.N.: Generating links by mining quotations. In: Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, New York, NY, USA (2008)
25. Lopresti, D.: Performance evaluation for text processing of noisy inputs. In: SAC'05: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 759–763. ACM, New York, NY, USA (2005)
26. Miller, S., Fox, H., Ramshaw, L., Weischedel, R.: A novel use of statistical parsing to extract information from text. In: Proceedings of the 1st Conference on North American Chapter of the ACL, pp. 226–233. Seattle, USA (2000)
27. Moschitti, A.: Kernel methods, syntax and semantics for relational text categorization. In: CIKM'08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 253–262. New York, NY, USA, ACM (2008)
28. Moschitti, A., Cosmin, A.B.: A semantic kernel for predicate argument classification. In: CoNLL-2004, Boston, MA, USA (2004)
29. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Comput. Linguist., Special Issue on Semantic Role Labeling (3):245–288 (2008)
30. Nguyen, T., Moschitti, A., Riccardi, G.: Convolution kernels on constituent, dependency and sequential structures for relation extraction. In: Proceedings of EMNLP (2009)
31. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
32. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. **5**, 101–141 (2004)
33. Roth, D., tau Yih, W.: Probabilistic reasoning for entity and relation recognition. In: Proceedings of the COLING-2002, Taipei, Taiwan (2002)
34. Sanderson, R., Watry, P.: Integrating data and text mining processes for digital library applications. In: Proceedings of JCDL 2007, New York, NY, USA (2007)
35. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
36. Subramaniam, L.V., Roy, S., Faruquie, T.A., Negi, S.: A survey of types of text noise and techniques to handle noisy text. In: AND'09: Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data, pp. 115–122. New York, NY, USA, ACM (2009)
37. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
38. Wang, M.: A re-examination of dependency path kernels for relation extraction. In: Proceedings of the 3rd International Joint Conference on Natural Language Processing-IJCNLP (2008)
39. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
40. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3**, 1083–1106 (2003)
41. Zhang, M., Su, J., Wang, D., Zhou, G., Tan, C.L.: Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In: Proceedings of IJCNLP'2005, Lecture Notes in Computer Science (LNCS 3651), pp. 378–389. Jeju Island, South Korea (2005)
42. Zhang, M., Zhang, J., Su, J., Zhou, G.: A composite kernel to extract relations between entities with both flat and structured features. In: Proceedings of COLING-ACL 2006, pp. 825–832. (2006)
43. Zhang, M., Zhou, G., Aw, A.: Exploring syntactic structured features over parse trees for relation extraction using kernel methods. Inf. Process. Manage. **44**(2), 825–832 (2006)
44. Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. In: Proceedings of the 43rd Meeting of the ACL, pp. 419–426. Ann Arbor, MI, USA (2005)
45. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd Meeting of the ACL, pp. 427–434. Ann Arbor, USA, June 2005
46. Zhou, G., Zhang, M., Ji, D., Zhu, Q.: Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: Proceedings of EMNLP-CoNLL 2007, pp. 728–736. (2007)
47. Zhou, X., Han, H., Chankai, I., Prestrud, A., Brooks, A.: Approaches to text mining for clinical medical records. In: Proceedings of SAC 2006, New York, NY, USA (2006)
48. Zhou, X., Pan, X., Ren, Y.: Web mining of relations from xml and construct database schema. In: CIMCA'06: Proceedings of CIMCA 2006, Washington, DC, USA (2006)