Studies in Computational Intelligence 589

Roberto Basili
Cristina Bosco
Rodolfo Delmonte
Alessandro Moschitti
Maria Simi *Editors*

Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project

Springer

Studies in Computational Intelligence

Volume 589

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Roberto Basili · Cristina Bosco
Rodolfo Delmonte · Alessandro Moschitti
Maria Simi
Editors

Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project

Springer

Editors
Roberto Basili
Department of Computer Science, Systems and Production
University of Rome Tor Vergata
Rome
Italy

Cristina Bosco
Department of Computer Science
University of Turin
Turin
Italy

Rodolfo Delmonte
Department of Language and Cultural Studies, Department of Computer Science
Ca' Foscari University of Venice
Venezia
Italy

Alessandro Moschitti
Department of Computer Science and Information Engineering
University of Trento
Trento
Italy

Maria Simi
Department of Computer Science
University of Pisa
Pisa
Italy

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-14205-0 ISBN 978-3-319-14206-7 (eBook)
DOI 10.1007/978-3-319-14206-7

Library of Congress Control Number: 2014958283

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

Portale per l'Accesso alle Risorse in Lingua Italiana (PARLI) is a project partially funded by the Ministero Italiano per l'Università e la Ricerca (PRIN 2008) from 2008 to 2012. The project was proposed by research units working in seven Italian universities, namely the University of Torino with a subunit at the University of Napoli "Federico II", the University of Pisa, the University of Roma "Tor Vergata", the University of Trento, the University of Venezia "Ca' Foscari". Moreover the Fondazione Bruno Kessler (FBK, Trento), the Istituto di Linguistica Computazionale "Antonio Zampolli"—CNR (Pisa) and the Associazione Italiana per l'Intelligenza Artificiale (AI*IA) played in the project the role of cooperating partners.

As the title of the project itself shows, PARLI mainly aimed at monitoring and fostering the harmonic growth and coordination of the activities of Italian NLP. In addition to that, it also proposes itself as a point of reference for the development of Italian NLP. According to this perspective, a web portal (http://parli.di.unito.it/) has been developed as a reference point for Italian NLP and for monitoring related activities. It includes links to existing resources and tools developed for Italian or applied to it. It mainly benefits from the data made available within the Evalita evaluation campaigns (http://www.evalita.it/) held in 2007, 2009 and 2011, and is linked by the NLP section of the AI*IA website (http://www.aixia.it/).

As far as the harmonic growth and coordination of Italian NLP is concerned, several activities promoted by PARLI members are attested by more than 50 publications, issued within the project, in international conferences, journals and workshops, among which are those related to the Evalita experiences, which were mainly organized and intensively participated by the PARLI members and cooperating partners.

There are several directions in which research on NLP has made considerable progress in the last few years. The chapters collected in this volume are selected as a sample of those performed for Italian NLP and especially oriented to the goals of the PARLI project, namely the consolidation and harmonization of existing linguistic resources, the development of new resources and tools that can harmonically operate and grow together, and the study of models for the comparison and evaluation of tools and resources.

Even if more and more treebanks are currently available also for lesser studied languages, none of the existing resources for Italian is large enough to train and test NLP systems with high reliability. This is also because they are featured by annotations which are far from standards applied in larger and well-known data sets. The consolidation and harmonization of these existing linguistic resources is at issue in the article of Simi, Montemagni and Bosco, where a methodology for merging and converting treebanks in a standard annotation format is designed. The format is applied to two existing Italian resources, i.e. Turin University Treebank (TUT) and ISST–TANL, in order to build a larger data set in the standard de facto Stanford Dependency format. Also, the contribution of Delmonte refers to issues related to standards for annotation. It highlights a peculiar limit of the formats of resources on which state-of-the-art parsers are currently trained, i.e. the exclusion of null elements, and faces the problems derived from the conversion in a format almost semantically complete which includes null elements.

The development of new resources that can grow and cooperate together is the topic of the contribution of Sanguinetti, Lesmo and Bosco, where a recently released parallel treebank is proposed for cross-linguistic comparisons among Italian, English and French, and a study for the development of a dependency-based alignment system. This resource applies the same format of the TUT and takes advantage of the tools developed for this treebank; in addition, it can influence machine translation as well as linguistic investigations. Another kind of approach is taken in the chapter by Magnini, Zanoli and Firoj, which presents a comparative analysis of named entities extraction from both written and spoken documents, thus introducing a new perspective related to spoken language.

The contributions of Croce, Basili and Moschitti and the that by Croce, Filice and Basili describe the development of tools and related methodologies. The former chapter tackles the definition and evaluation of the semantically Smoothed Partial Tree Kernel, which is a generalized formulation of one of the most performant Convolution Kernels, i.e. the Tree Kernel, by extending the similarity between tree structures with node similarities. The latter chapter instead discusses a perspective centred on Convolution Kernels and the formulation of a Partial Tree Kernel that integrates syntactic information and lexical generalization, in order to define methods able to express the meaning of phrases or sentences as operations on lexical representations.

The contribution of Alicante, Bosco, Corazza and Lavelli and the that by Mazzei deal with the study of models for comparison and evaluation of tools and resources. The former chapter is a collection of parsing experiments performed on TUT data in order to compare the two main paradigms, i.e. dependency and constituency, and forms of annotation featured by a different amount of linguistic knowledge. In the chapter by Mazzei, instead, an ensemble system for dependency parsing of Italian is presented where three parsers known in the literature are separately trained and combined by means of a majority vote on a common data set.

According to the spirit of the project PARLI, the resources and tools created within the project or made available by their partners are freely distributed. Moreover, as attested also by the richness of the future directions drawn in the

chapters here collected, it should be desirable that the activities associated with PARLI do not terminate at the end of the funded project itself. PARLI, the portal and the resources associated with it should continue to be managed even later, hoping they could be a key factor in resource development in computational linguistics for Italian and beyond.

Roberto Basili
Cristina Bosco
Rodolfo Delmonte
Alessandro Moschitti
Maria Simi

Acknowledgment

We all thank and remember Leonardo Lesmo, coordinator of the PARLI project and valuable teacher, colleague and friend.

Contents

Contributors

Firoj Alam SIS Lab, Department of Information Engineering and Computer Science, University of Trento, Povo (TN), Italy

Anita Alicante Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università di Napoli Federico II, Naples, Italy

Roberto Basili Department of Computer Science, Systems and Production, University of Roma Tor Vergata, Rome, Italy

Cristina Bosco Dipartimento di Informatica, Università di Torino, Torino, Italy

Anna Corazza Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università di Napoli Federico II, Naples, Italy

Danilo Croce Department of Computer Science, Systems and Production, University of Roma Tor Vergata, Rome, Italy

Rodolfo Delmonte Department of Language and Cultural Studies, Department of Computer Science, Ca' Foscari University of Venice, Venezia, Italy

Simone Filice Department of Computer Science, Systems and Production, University of Roma Tor Vergata, Rome, Italy

Alberto Lavelli HLT Research Unit, Fondazione Bruno Kessler, Povo, TN, Italy

Bernardo Magnini FBK-irst, Povo (TN), Italy

Alessandro Mazzei Dipartimento di Informatica, Università di Torino, Torino, Italy

Simonetta Montemagni Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR), Pisa, Italy

Alessandro Moschitti Department of Computer Science and Information Engineering, University of Trento, Povo (TN), Italy

Manuela Sanguinetti Dipartimento di Informatica, Università di Torino, Torino, Italy

Maria Simi Dipartimento di Informatica, Università di Pisa, Pisa, Italy

Roberto Zanoli FBK-irst, Povo (TN), Italy

Part I
Linguistic Resources

Harmonizing and Merging Italian Treebanks: Towards a *Merged Italian Dependency Treebank* and Beyond

Maria Simi, Simonetta Montemagni and Cristina Bosco

Abstract In this paper we address the challenge of combining existing CoNLL-compliant dependency-annotated corpora with the final aim of constructing a bigger treebank for the Italian language. To this end, we defined a methodology for mapping different annotation schemes, based on: (i) The analysis of similarities and differences of considered source and target dependency annotation schemes; (ii) The analysis of the performance of state of the art dependency parsers trained on the source and target treebanks; (iii) The mapping of the source annotation scheme(s) onto a set of target (possibly underspecified) data categories. This methodology was applied in two different case studies. The first one was aimed at constructing a "Merged Italian Dependency Treebank" (MIDT) starting from existing Italian dependency treebanks, namely TUT and ISST–TANL. The second case study, still ongoing, consists in the conversion of the MIDT resource into the Stanford Dependencies *de facto* standard with the final aim of developing an "Italian Stanford Dependency Treebank" (ISDT).

Keywords Treebank · Italian · Harmonization and merging of resources

M. Simi
Dipartimento di Informatica, Università di Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy
e-mail: simi@unipi.it

S. Montemagni
Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR),
Via G. Moruzzi 1, 56124 Pisa, Italy
e-mail: simonetta.montemagni@ilc.cnr.it

C. Bosco (✉)
Dipartimento di Informatica, Università di Torino,
Corso Svizzera 185, 10149 Torino, Italy
e-mail: bosco@di.unito.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_1

1 Introduction

The limited availability of training resources is a widely acknowledged bottleneck for machine learning approaches for Natural Language Processing (NLP). This is also the case of dependency treebanks within statistical dependency parsing. If on the one hand a small size resource used for training doesn't guarantee reliable results, on the other hand the development of a bigger treebank is a very expensive and time-consuming task. This state of affairs motivates the current trend towards the harmonization and merging of existing data sets, possibly converting them into *de facto* standards.

Within this general picture, let us consider the case of Italian. For this language there are four available dependency treebanks. Three of them were developed by national research institutions: the Turin University Treebank (TUT)[1] developed by the NLP group of the University of Turin [1]; the treebank called ISST–TANL, which was developed as a joint effort by the Istituto di Linguistica Computazionale (ILC–CNR) and the University of Pisa and originating from the Italian Syntactic-Semantic Treebank or ISST [2]; the Venice Italian Treebank (VIT) developed by the University Ca' Foscari of Venice [3]. A further Italian dependency treebank was developed in the framework of an international project, the Copenhagen Dependency Treebank [4]. Interesting to note, each of these resources, independently developed applying different dependency-based annotation schemes, has a quite limited size, ranging from around 94,000 tokens of TUT to about 60,000 tokens of the Italian CDT section.

In spite of their limited size, some of these resources have successfully been used for training and/or evaluating dependency parsing systems. For instance, TUT was repeatedly used within the parsing task of the EVALITA evaluation campaign[2] in 2007, 2009 and 2011, for both training and testing dependency and constituency parsing systems. A previous version of ISST–TANL, namely ISST–CoNLL, was used for the CoNLL-2007 Shared Task on multilingual dependency parsing as far as Italian is concerned [5, 6]. ISST–TANL was used in EVALITA 2009 and 2011 for two different tasks, syntactic parsing [7] and domain adaptation [8] respectively, and also in the SPLeT 2012 Shared Task on Dependency Parsing of Legal Texts [9].[3] For these resources to be used in the framework of international evaluation campaigns, preliminary steps towards their harmonization and merging were performed. First, the native annotation formats have been converted into the CoNLL representation standard. Second, within EVALITA 2009 a common set of test data was annotated following the TUT and ISST–TANL annotation guidelines, thus making it possible to start investigating the influence of the design of treebanks on the output of participating parsing systems [10].

[1] http://www.di.unito.it/~tutreeb.

[2] http://www.evalita.it/.

[3] http://poesix1.ilc.cnr.it/splet_shared_task/.

Despite the encouraging results achieved by exploiting these treebanks in the above mentioned initiatives, we are aware that the relatively small size of these resources makes them usable in a restricted variety of tasks with an impact on the reliability of achieved results. By contrast, the availability of a larger treebank, harmonizing and merging the original annotated resources, should result in crucial advancements for the Italian NLP. Such an effort would be even more promising if the harmonization and merging could be carried out with respect to a *de facto* standard for comparing treebanks at the dependency level: as originally claimed by [11], this is currently represented by the Stanford Dependencies (henceforth SD) annotation scheme [12].

The question at this point is whether the harmonization and merging of existing treebanks is nowadays a realistic goal. Since the early 1990s, different initiatives have been devoted to the definition of standards for the linguistic annotation of corpora with a specific view to re-using and merging existing annotated resources. The starting point was represented by the EAGLES (Expert Advisory Groups on Language Engineering Standards) initiative, which ended up with providing provisional standard guidelines [13], operating at the level of both content (i.e. the linguistic categories) and encoding format. More recent initiatives, e.g. LAF/GrAF [14, 15] and SynAF [16] representing on-going ISO TC37/SC4 standardization activities[4], rather focused on the definition of a pivot format capable of representing diverse annotation types of varying complexity without providing specifications for the annotation of content categories (i.e., the labels describing the associated linguistic phenomena), for which standardization appeared since the beginning to be a much trickier matter. Recently, other standardization efforts such as ISOCat [17] tackled this latter issue by providing a set of data categories at various levels of granularity, each accompanied by a precise definition of its linguistic meaning. Unfortunately, the set of dependency categories within ISOCat is still basic and restricted, and for this reason it cannot be used for harmonizing and merging real treebanks.

The work illustrated in this paper is concerned with the harmonization and merging of CoNLL-compliant dependency-annotated corpora, with a specific view to content categories. Since standardization of content categories is still at an early development stage, we could not rely on it. Therefore, to address the challenge of combining existing treebank resources with the final aim of constructing a bigger treebank for the Italian language we defined a methodology for translating between different annotation schemes and merging them, articulated into the following steps: (i) analysis of similarities and differences of considered source and target dependency annotation schemes; (ii) analysis of the performance of state of the art dependency parsers trained on source and target treebanks; (iii) mapping of the source annotation scheme(s) onto a set of target sets of (possibly underspecified) data categories. This methodology was applied in two different case studies.

The first case study was carried out within the national project "Portal for the Access to the Linguistic Resources for Italian" (PARLI), where an annotation scheme to be used as a "bridge" between the native schemes was defined and used for the

[4] http://www.tc37sc4.org/.

harmonization and merging of the TUT and ISST–TANL resources. This resulted in the construction of the *Merged Italian Dependency Treebank* (MIDT). The second case study, performed in the framework of a collaboration with Google, consists in the conversion of the resource resulting from the first case study, i.e. MIDT, into the Stanford Dependency *de facto* standard. The MIDT_to_SD conversion process, described in [18], generates a new standard-compliant resource, i.e. the *Italian Stanford Dependency Treebank* (or ISDT).

The paper is organised as follows. Sections 2, 3 and 4 illustrate step by step the first case study which resulted in the construction of the MIDT resource. Finally, Sect. 5 reports preliminary results of the second case study.

2 The TUT and ISST–TANL Treebanks

The TUT and ISST–TANL resources differ under different respects, at the level of both corpus composition and adopted representations.

For what concerns size and composition, TUT currently includes 3,452 Italian sentences (i.e. 102,150 tokens in TUT native, and 93,987 in CoNLL[5]) and represents five different text genres (newspapers, Italian Civil Law Code, JRC-Acquis Corpus,[6] Wikipedia and the *Costituzione Italiana*). ISST–TANL includes instead 3,109 sentences (71,285 tokens in CoNLL format), which were extracted from the "balanced" ISST partition [2] exemplifying general language usage and consisting of articles from newspapers and periodicals, selected to cover a high variety of topics (politics, economy, culture, science, health, sport, leisure, etc.).

As far as annotation is concerned, both TUT and ISST–TANL schemes belong to the dependency paradigm. TUT applies the major principles of the dependency grammar [19] using a rich set of grammatical relations, and in the native version it includes null elements to deal with non-projective structures, long distance dependencies, equi phenomena, pro drop and elliptical structures.[7] The ISST–TANL annotation scheme originates from FAME [20], an annotation scheme which was developed starting from *de facto* standards and which was specifically conceived for complying with the basic requirements of parsing evaluation, and—later—for the annotation of unrestricted Italian texts. Note, however, that in what follows we will refer to the CoNLL-compliant versions of these resources which, in spite of their sharing the same representation format, still differ significantly, e.g. they assume different inventories of dependency types characterized by different degrees of granularity in the representation of specific relations or they adopt different annotation criteria even when the dependency type appears to be the same.

[5] In the following we will refer only to number of tokens in CoNLL format.

[6] http://langtech.jrc.it/JRC-Acquis.html.

[7] The CoNLL format does not include null elements, but the projectivity constraint is maintained at the cost of a loss of information with respect to native TUT in some cases).

2.1 Comparing the Annotation Schemes

In spite of the fact that TUT and ISST–TANL annotations fall within the same broader family of schemes, there are significant differences which make the harmonization and merging of the two resources quite a challenging task. To put it in other words, if on the one hand there is a core of syntactic constructions for which the analysis given by different annotation schemes agree in all important respects, on the other hand there are also important differences concerning the inventory of dependency types and their linguistic interpretation, head selection criteria, the projectivity constraint as well as with respect to the analysis of specific syntactic constructions. In what follows, we summarize the main dimensions of variation with a specific view to the merging issues they arise.

Head selection criteria

Criteria for distinguishing the head and the dependent within dependency relations have been widely discussed in the linguistic literature, not only in the dependency grammar tradition, but also within other frameworks where the notion of syntactic head plays an important role. Unfortunately, different criteria have been proposed, some syntactic and some semantic, which do not lead to a single coherent notion of dependency [21]. Head selection thus represents an important and unavoidable dimension of variation between the TUT and ISST–TANL schemes, especially for what concerns constructions involving grammatical function words with respect to which there is no general consensus in the tradition of dependency grammar as to what should be regarded as the head and what should be regarded as the dependent. Let us focus on the following tricky cases: namely, the determiner–noun relation within nominal groups, the preposition–noun relation within prepositional phrases, the complementizer–verb relation in subordinate clauses as well as the auxiliary-main verb relation in complex verbal groups.

TUT always assigns heads on the basis of syntactic criteria, i.e. in all constructions involving one function word and one content word (e.g. determiner–noun, preposition–noun, complementizer–verb) the head role is always played by the function word. The only exception is represented by auxiliary-main verb constructions where the head role is played by the main verb.

By contrast, in ISST–TANL head selection follows from a combination of syntactic and semantic criteria: i.e. whereas in the determiner–noun and auxiliary-verb constructions the head role is assigned to the semantic head (noun/verb), in preposition–noun and complementizer–verb constructions the head role is played by the element which is subcategorized for by the governing head, i.e. the preposition and the complementizer.

Note that this different strategy in the head selection explains the asymmetric treatment of determiner–noun constructions with respect to the preposition–noun

ones in ISST–TANL and the fact the for TUT the same dependency type is used for both cases.

Granularity and inventory of dependency types

TUT and ISST–TANL annotation schemes assume different inventories of dependency types characterized by different degrees of granularity in the representation of specific relations. The different degree of granularity of the annotation schemes is testified by the size of the adopted dependency tagsets, including 72 dependency types in the case of TUT and 29 in the case of ISST–TANL. Interestingly however, it is not always the case that the finer grained annotation scheme—i.e. TUT—is the one providing more granular distinctions: whereas this is typically the case, there are also cases in which more granular distinctions are adopted in the ISST-TANL annotation scheme. In what follows, we provide examples of both cases.

Consider first TUT relational distinctions which are neutralized at the level of ISST–TANL annotation. A difference in terms of granularity refers e.g. to the annotation of appositive (or unrestrictive) modifiers, which in TUT are annotated by resorting to a specific relation (`APPOSITION`), and which in ISST–TANL are not distinguished from other kinds of modifiers (`mod`). Similarly, TUT partitions predicative complements into two classes, i.e. subject and object predicative complements (`PREDCOMPL+SUBJ` and `PREDCOMPL+OBJ` respectively), depending on whether the complement refers to the subject or the object of the head verb.

Let us consider now the reverse case, i.e. in which ISST–TANL adopts finer-grained distinctions with respect to TUT: for instance, ISST–TANL envisages two different relation types for determiner–noun and preposition–noun constructions (`det` and `prep` respectively), whereas TUT represents both cases in terms of the same relation type (`ARG`). This latter example follows from another important dimension of variation between the two schemes, concerning head selection.

Another interesting and more complex example can be found for what concerns the partitioning of the space of prepositional complements, be they modifiers or subcategorized arguments. TUT distinguishes between `modifier`(s) on the one hand and subcategorised arguments on the other hand; the latter are further distinguished between indirect objects (`INDOBJ`) and all other types of indirect complements (`INDCOMPL`). ISST–TANL neutralizes such a distinction by resorting to a single dependency type, i.e. `comp` (mnemonic for complement), for all relations holding between a head and a prepositional complement, whether a modifier or a subcategorized argument. On the other hand, `comp`(lements) are further subdivided into semantically oriented categories, such as temporal, locative or indirect complements (`comp_temp`, `comp_loc` and `comp_ind`).

Same dependency type, different annotation criteria

Even when—at first glance—the two schemes show common dependency types, they can diverge at the level of their interpretation, and thus of the underlying annotation criteria. This is the case, for instance, of the “object” relation which in the TUT

annotation scheme refers to the direct argument (either in the nominal or clausal form) occurring at least and most once and expressing the subcategorized object, and which in ISST–TANL is meant to denote the relation holding between a verbal head and its non-clausal direct object (other dependency types are foreseen to mark clausal complements).

Another interesting example is represented by relative clauses. TUT and ISST–TANL follow the same strategy in the representation of standard relative clauses, according to which the head of the relative clause is the verb and the relative pronoun is governed by it as a standard argument. The verbal head is then connected to the antecedent noun (i.e. the noun governing the clause) through a specific relation, `RELCL` in TUT and `mod_rel` in ISST–TANL. However, TUT also treats so-called reduced relative clauses, i.e. constructions where there is no overt relative pronoun and the verb appears in the participial form (either present or past participle), in the same way; namely, by using the same relation type to link the verb of the reduced relative clause to the governing noun. In ISST–TANL, constructions without overt relative pronouns are instead represented by resorting to a general modifier relation (`mod`).

Projectivity of dependency representations

Projectivity is an important constraint in dependency grammar, relating dependency structures to linear realizations. If on the one hand most NLP systems for dependency parsing assume projectivity, on the other hand this is not the case on the linguistic side where non-projective are recognized in specific linguistic constructions (e.g. long-distance dependencies) mainly occurring in flexible word order languages (such as Italian). Whereas ISST–TANL corpus allows for non-projective representations, TUT assumes the projectivity constraint.

Treatment of specific constructions

Further important differences between TUT and ISST–TANL annotation schemes are concerned with the treatment of coordination and punctuation, phenomena which are particularly problematic to deal with in the dependency framework.

Besides the general issue widely discussed in the literature of whether coordination can be analyzed in terms of binary asymmetrical relations holding between a head and a dependent, there are different ways put forward to deal with it. In both TUT and ISST–TANL resources, coordinated constructions are considered as asymmetric structures with a main difference: while in ISST–TANL the conjunction and the subsequent conjuncts are all linked to the first conjunct, in TUT the conjuncts starting from the second one are linked to the immediately preceding conjunction.

Also the treatment of punctuation is quite problematic in the framework of a dependency annotation scheme, although this has not been specifically dealt with in the linguistic literature. Both TUT and ISST–TANL schemes cover punctuation with main differences holding at the level of both dependency types and head selection criteria. Whereas ISST–TANL has just one dependency type for all punctuation tokens,

TUT distinguishes different dependency types depending on the involved punctuation token and syntactic construction. For example, in TUT an explicit notion of parenthetical is marked while in ISST–TANL it is not. Significant differences also lie at the level of the head assignment criteria: in TUT the head of the punctuation tokens in the parenthetical structure coincides with the governing head of the subtree covering the parenthetical structure (i.e. it is external to the parenthetical structure), whereas in ISST–TANL the paired punctuation marks of the parenthetical structure are both connected to the head of the delimited phrase (i.e. internally to the parenthetical).

Another important difference concerns the sentence root: TUT annotation scheme enforces the single root constraint, whereas this does not hold in the case of ISST–TANL where multiple-rooted sentences can in principle occur. Other important differences holding between TUT and ISST–TANL are concerned with sentence splitting, tokenization and morpho-syntactic annotation with an impact at the level of dependency annotation. For the specific concerns of this paper focusing on the merging of dependency annotations, we won't further discuss these aspects which represent however important prerequisites for the merging of dependency annotations.

3 TUT and ISST–TANL as Training Corpora

In [10], a dependency-based analysis of the performance of state of the art parsers participating in EVALITA 2009 (two stochastic parsers and a rule-based one) with respect to a shared test set was reported, with the final aim of assessing the impact of annotation schemes on parsing results. In particular, for each relation in the TUT and ISST–TANL dependency annotation schemes, the performance of the three parsers was analyzed in terms of Precision (P), Recall (R) and related f-score. In order to identify problematic areas of parsing, both TUT and ISST–TANL dependency-relations were partitioned into three classes (i.e. low-, medium- and best-scored dependency relations) with respect to the associated f-score, which was taken to reflect their parsing difficulty (for more details see [10]). Achieved results showed that the improvement of parsing technology should proceed hand in hand with the development of more suitable representations for annotated syntactic data. In this paper we are dealing with the latter issue: we believe that the results of this comparative analysis should also be taken into account in the definition of the merging methodology.

Similar trends were observed in the performance of parsers against TUT and ISST–TANL. First, in both cases hard to parse relations include "semantically loaded" relations such as `comp_temp`, `comp_loc` and `comp_ind` for ISST–TANL and `APPOSITION` and `INDOBJ` for TUT. Moreover, relations involving punctuation appeared to be difficult to parse for statistical parsers in the case of TUT, whereas the rule-based parser had problems dealing with coordinate structures in ISST–TANL; it should be noted however that ISST–TANL `con`/`conj` relations show values very close to the low threshold value also in the case of the stochastic

parsers. This contrastive analysis thus confirmed a widely acknowledged claim, i.e. that coordination and punctuation phenomena still represent particularly challenging areas for parsing [22]. The problems raised by the analysis of "semantically loaded" relations in the case of both treebanks suggest that the parsers do not appear to have sufficient evidence to deal reliably with them; in principle, the solutions to the problem range from increasing the size of the training corpus, to neutralizing their distinction at this annotation level and postponing their treatment to further processing levels.

Concerning the best scored relations, it came out that in both cases they mainly refer to "local" relations. Interesting to note, there is a significant overlapping between the two sets: e.g. the TUT `ARG` and the ISST–TANL `det/prep` together have the same coverage; the same holds for the TUT `AUX+PASSIVE/AUX+TENSE` relations with respect to the ISST–TANL `aux` relation.

4 Merging TUT and ISST–TANL

In this section, we illustrate the work done towards merging the two annotated resources, by defining a bridge annotation scheme to be used as an interlingua for converting the individual treebanks and combining them into a wider resource. Whereas we are aware of previous efforts of combining different annotation types (e.g. ISOTimeML, PropBank, and FrameNet annotations as reported in [23]) as well as dependency structures of different languages (e.g. English vs. Japanese as discussed in [24]), to our knowledge this represents the first merging effort carried out with respect to different dependency annotation schemes defined for the same language: we will refer to them as dependency annotation "dialects".

In what follows, we first illustrate the criteria which guided the definition of a bridge annotation scheme to be used for merging the two resources (Sect. 4.1); second, in order to test the adequacy of the resulting annotation scheme as far as dependency parsing is concerned we report the parsing results achieved by exploiting the MIDT resources as training data (Sect. 4.2).

4.1 Defining a Bridge Annotation Scheme for MIDT

The results of the comparative analysis detailed in Sect. 2.1 are summarized in columns 2, 3 and 4 of Table 1, where for each relation type in a given scheme the corresponding relation(s) are provided as far as the other scheme is concerned. The fourth column (headed "DIFF") provides additional information for what concerns the type of correspondence holding between ISST–TANL and TUT dependency categories: two different values are foreseen, which can also be combined together, corresponding to whether the correspondence involves different head selection criteria ("Hsel")

Table 1 ISST–TANL, TUT and MIDT linguistic ontologies

ID	ISST–TANL	TUT	DIFF	MIDT
1	ROOT	TOP		_ROOT
2	arg	No equivalent relation (see 5, 21)	covg	_ARG
3	aux	AUX(+PASSIVE +PROGRESSIVE +TENSE)		_AUX
4	clit	EMPTYCOMPL SUBJ/SUBJ+IMPERS		_CLIT
5	comp	INDCOMPL SUBJ/INDCOMPL COORD+COMPAR	covg	_COMP
6	comp_ind	INDOBJ SUBJ/INDOBJ		_COMP
7	comp_loc	No equivalent relation (see 5)	covg	_COMP
8	comp_temp	No equivalent relation (see 5)	covg	_COMP
9	con	COORD(+BASE +ADVERS +COMPAR +COND +CORRELAT +ESPLIC +RANGE +SYMMETRIC)	covg Hsel	_COORD
10	concat	CONTIN(+LOCUT +DENOM +PREP)		_CONCAT
11	conj	COORD2ND(+BASE +ADVERS +COMPAR +COND +CORRELAT +ESPLIC) COORDANTEC+CORRELAT	covg Hsel	_COOR2ND
12	det	ARG	Hsel	_DET, _ARG
13	dis	No equivalent relation (see 9)	covg	_COORD
14	disj	No equivalent relation (see 11)	covg	_COOR2ND
15	mod	APPOSITION RMOD RMOD+RELCL+REDUC INTERJECTION COORDANTEC+COMPAR	covg	_MOD
16	mod_loc	No equivalent relation (see 15)	covg	_MOD
17	mod_rel	RMOD+RELCL		_RELCL
18	mod_temp	No equivalent relation (see 15)	covg	_MOD
19	modal	No equivalent relation (see 3)	Hsel covg	_AUX
20	neg	No equivalent relation (see 15)	covg	_NEG
21	obj	OBJ SUBJ/OBJ EXTRAOBJ	covg	_OBJ
22	pred	PREDCOMPL(+SUBJ +OBJ) RMODPRED(+OBJ +SUBJ)		_PRED
23	pred_loc	No equivalent relation (see 22)	covg	_PRED
24	pred_temp	No equivalent relation (see 22)	covg	_PRED
25	prep	ARG	Hsel	_PREP, _ARG
26	punc	CLOSE(+PARENTHETICAL +QUOTES) END INITIATOR OPEN(+PARENTHETICAL +QUOTES) SEPARATOR		_PUNC
27	sub	ARG	Hsel	_SUB, _ARG
28	subj	SUBJ EXTRASUBJ	covg	_SUBJ
29	subj_pass	OBJ/SUBJ		_SUBJ

and/or a different linguistic interpretation resulting in a different coverage ("covg"). It can be noted that the emerging situation is quite heterogeneous.

The only simple cases are represented by (a) the root, relative clause and passive subject cases for which we observe a 1:1 mapping, and (b) the relation(s) involving auxiliaries in complex tense constructions characterized by a 1:n mapping. As far as (b) is concerned, in principle the TUT relation distinctions might be recovered by also taking into account the lexical and morpho-syntactic features associated with the involved auxiliary and main verbal tokens. In both (a) and (b) cases, however, the identification of a bridge category to be used for merging purposes does not appear to be problematic at all (see below).

A slighly more complex case is represented by the determiner–noun, preposition–noun and complementizer–verb relations whose treatment in the two annotation schemes is different both at the level of involved relation types and head selection criteria. For these cases, the merging process should also be able to deal with the "external" consequences at the level of the overall tree structure as far as the attachment of these constructions is concerned. For instance, depending on the scheme, in a sentence like *I read the book* the object of reading would be either the article (TUT) or the noun (ISST–TANL). In these cases, besides defining a semantically coherent bridge category compatible with both TUT and ISST–TANL annotations, the conversion process is not circumscribed to the dependency being converted but should also deal with the restructuring of the sub-tree whose head governs the dependency head.

Most of remaining dependency relations involve different, sometimes orthogonal, sets of criteria for their assignment and are therefore more difficult to deal with for merging purposes. Consider, as an example, the direct object relation, already discussed in Sect. 2.1: in ISST–TANL the relation `obj` is restricted to non-clausal objects (typically in nominal form), whereas the TUT `OBJ` relation also includes clausal ones. This difference in terms of coverage follows from the fact that whereas TUT implements a pure dependency annotation where the dependency type does not vary depending on the complement type (e.g. clausal vs. nominal objects), in ISST–TANL all clausal complements are treated under a specific relation type, named `arg`. This represents a much trickier case to deal with for merging purposes: here it is not a matter of choosing between two different representation strategies, but rather of converging on a possibly underspecified representation type which could be automatically reconstructed from both TUT and ISST–TANL resources. If on the one hand in TUT it is possible to recover the ISST–TANL notion of `arg` by exploiting the morpho-syntactic features of the tokens involved in the relation, on the other hand it is impossible to automatically recover the TUT notion of OBJ starting from ISST–TANL annotation only (in this case information about the subcategorization properties of individual verbs would be needed).

Another problematic conversion area is concerned with the representation of deverbal nouns (e.g. *destruction*) whose annotation in TUT is carried out in terms of the underlying predicate-argument structure (i.e. by marking relations such as subject, object, etc.) whereas in ISST–TANL is marked by resorting to generic surface (e.g. `comp`(lement)) relations. As in the subordination case, the only possible solution here is to converge on a representation type which can be automatically

reconstructed from both TUT and ISST–TANL resources by combining morpho-syntactic and dependency information.

It should also be noted that there are semantically-oriented distinctions which are part of the ISST–TANL annotation scheme (e.g. temporal and locative modifiers, i.e. `mod_temp` vs. `mod_loc`) but which do not find a counterpart in the CoNLL version of the TUT treebank. In this case the only possible solution consists in neutralizing such a distinction at the level of the MIDT representation.

The conversion process had also to deal with cases for which the difference was only at the level of annotation criteria rather than of the dependency types. Consider for instance the treatment of coordination phenomena. Both TUT and ISST–TANL foresee two different relations, one for linking the conjunction with one of the conjuncts (i.e. the ISST–TANL `con` and the TUT `COORD` relations) and the other one for connecting the conjoined elements (i.e. the ISST–TANL `conj` and the TUT `COORD2ND` relations). In spite of this parallelism at the tagset level, the strategy adopted for representing coordinate structures is different in the two resources: whereas ISST–TANL takes the first conjunct as the head of the whole coordinate structure and all subsequent conjoined elements and conjunctions are attached to it, in TUT both the conjunction and the conjunct are governed by the element immediately preceding it. In this case the conversion towards MIDT consists in restructuring the internal structure of the coordinate structure.

For each set of corresponding ISST–TANL and TUT categories, the last column of Table 1 contains the MIDT counterpart. The definition of the MIDT dependency tagset was mainly guided by practical considerations: namely, bridge categories should be automatically reconstructed by exploiting morpho-syntactic and dependency information contained in the original ISST–TANL and TUT resources. In MIDT, we also decided to neutralize semantically-oriented distinctions (such as the subject of passive constructions, or the indirect object) which turned out to be problematic (see Sect. 3) to be reliably identified in parsing in spite of their being explicitly encoded in both annotation schemes. Last but not least, the resulting MIDT tagset was also compared with *de facto* dependency annotation standards adopted for different languages: among them it is worth mentioning here the annotation tagsets proposed by syntactic annotation initiatives like TIGER, ISST, Sparkle and EAGLES as reported in [16] or the most recent Stanford typed dependencies representation [25].

It should be noted that, in some cases, MIDT provides two different options, corresponding to the TUT and ISST–TANL styles for dealing with the same construction: this is the case of determiner–noun, preposition–noun, complementizer–verb and auxiliary-main verb relations whose MIDT representation is parameterizable: for the time being only one possible option has been activated. The final MIDT tagset contains 21 dependency tags (as opposed to the 72 tags of TUT and the 29 of ISST–TANL), including the different options provided for the same type of construction. The question at this point is whether the MIDT annotation scheme is informative enough and at the same time fully predictable to reliably be used for different purposes: in the following section a first though preliminary answer to this question is provided.

4.2 Using MIDT as Training Corpus

In this section we report the results achieved by using MIDT resources for training a dependency parsing system. We used DeSR (Dependency Shift Reduce), a transition-based statistical parser [26] which builds dependency trees while scanning a sentence and applying at each step a proper parsing action selected through a classifier based on a set of representative features of the current parse state. Parsing is performed bottom-up in a classical Shift/Reduce style, except that the parsing rules are special and allow parsing to be performed deterministically in a single pass. It is possible to specify, through a configuration file, the set of features to use (e.g. POS tag, lemma, morphological features) and the classification algorithm (e.g. Multi-Layer Perceptron [27], Support Vector Machine, Maximum Entropy). In addition, the parser can be configured to run either in left-to-right or right-to-left word order. An effective use of DeSR is the Reverse Revision parser [28], a stacked parser which first runs in one direction, and then extracts hints from its output to feed another parser running in the opposite direction. All these options allow creating a number of different parser variants, all based on the same basic parsing algorithm. Further improvement can then be achieved by the technique of parser combination [28], using a greedy algorithm, which preserves the linear complexity of the individual parsers and often outperforms other more complex algorithms.

Let us start from the results achieved by this parser in the framework of the evaluation campaign Evalita 2011 with the original TUT and ISST–TANL datasets distributed in the framework of the "Dependency Parsing" [29] and "Domain Adaptation" [8] tracks respectively. Table 2 reports, in the first two rows, the values of Labeled Attachment Score (LAS) obtained with respect to the ISST–TANL and TUT datasets with the technique of parser combination: 82.09 versus 89.88 %. This result is in line with what reported in [10], where a similar difference in performance was observed with respect to the TUT and ISST–TANL test sets: the composition of the training corpora and the adopted annotation schemes were identified as possible causes for such a difference in performance by the same parsers.

Table 2 Parsing results with native versus MIDT resources

TRAINING	TEST	PARSER	LAS (%)	LAS no punct
ISST–TANL_native_train	ISST–TANL_native_test	Parser comb.	82.09	Not available
TUT_native_train	TUT_native_test	Parser comb.	89.88	Not available
ISST–TANL_MIDT_train	ISST–TANL_MIDT_test	Best single	84.47	86.15 %
ISST–TANL_MIDT_train	ISST–TANL_MIDT_test	Parser comb.	84.99	86.78 %
TUT_MIDT_train	TUT_MIDT_test	Best single	89.23	90.74 %
TUT_MIDT_train	TUT_MIDT_test	Parser comb.	90.11	91.58 %
merged_MIDT_train	merged_MIDT_test	Best single	86.09	88.60 %
merged_MIDT_train	merged_MIDT_test	Parser comb.	86.66	89.04 %

The results reported in rows 3–6 have been obtained by training DeSR with the MIDT version of the TUT and ISST–TANL individual resources, whereas rows 7 and 8 refer to the merged MIDT resource. In all these cases two different LAS scores are reported, i.e. the overall score and the one computed by excluding punctuation. For the MIDT resources, the DeSR results achieved with the best single parser and the combination of parsers are reported. It can be noticed that in both cases an improvement is observed with respect to the native TUT and ISST–TANL resources, +0.23 and +2.76 % respectively. The last two rows refer to the results achieved with the merged resource used as training. The performance achieved by training the parser on the merged resource is still high, although lower than the result achieved with TUT_MIDT_train. The parsing model trained on the merged resource obtains the following results with respect to individual test sets: 83.43 % for ISST–TANL_MIDT_test and 88.03 % for TUT_MIDT_test, which represent slightly lower LAS scores than those obtained by using as training the corresponding resource. This shows that further harmonization and merging work might be required; however, achieved parsing results demonstrate that the resulting MIDT resource can effectively be used for training dependency parsers.

5 Beyond MIDT: Towards an *Italian Stanford Dependency Treebank*

In this section we report the results achieved so far in the second case study aimed at generating an *Italian Stanford Dependency Treebank* (or ISDT) starting from MIDT. To pursue this goal, the methodology defined for the harmonization and merging of existing treebaks was specialized for the conversion of MIDT into SD representation. Differently from the previous case, here the target annotation scheme is given: however, it may require specializations with respect to linguistic peculiarities of the language dealt with, namely Italian. The MIDT and SD annotation schemes are both dependency-based and therefore fall within the same broader family. This fact, however, does not guarantee per se an easy and linear conversion process from one to the other: as pointed out in the previous sections, the conversion of an annotation scheme can be quite a challenging task, even when this process is carried out within a same paradigm. In the case at hand, this task is made easier thanks to the fact that MIDT and SD schemes share similar design principles: for instance, in both cases preference is given to relations which are semantically contentful and useful to applications, or to relations linking content words rather than being indirectly mediated via function words (see design principles 2 and 5 respectively in [12]). Another peculiarity shared by MIDT and SD consists in the fact that they both neutralize the argument/adjunct distinction for what concerns prepositional complements, which is taken to be "largely useless in practice" as [12] claim. In what follows, we summarize the main dimensions of variation between MIDT and SD.

Consider first the granularity and inventory of dependency types. MIDT and SD annotation schemes assume different inventories of dependency types characterized by different degrees of granularity in the representation of specific relations. The different degree of granularity of the annotation schemes is testified by the size of the adopted dependency tagsets, including 21 dependency types in the case of MIDT and 48 in the case of SD. Interestingly however, it is not always the case that the finer grained annotation scheme—i.e. SD—is the one providing more granular distinctions: whereas this is typically the case, there are also cases in which more granular distinction are adopted in the MIDT annotation scheme.

Consider first SD relational distinctions which are neutralized at the level of the MIDT annotation. As reported in [12], so-called NP-internal relations are critical in real world applications: the SD scheme therefore includes many relations of this kind, e.g. `appos` (appositive modifier), `nn` (noun compound), `num` (numeric modifier), `number` (element of compound number) and `abbrev` (abbreviation). In MIDT all these relation types are lumped together under the general heading of `mod` (modifier). To deal with these cases, the MIDT to SD conversion has to simultaneously combine dependency and morpho-syntactic information (e.g. the morpho-syntactic category), which however might not be sufficient for recovering appositive modifiers for which further evidence is needed.

Let us consider now the reverse case, i.e. in which MIDT adopts finer-grained distinctions with respect to SD. For instance, MIDT envisages different relation types for auxiliary-verb and preposition–verb (within infinitive clauses, be they modifiers or subcategorized arguments) constructions, which are `aux` and `prep` respectively. By contrast, SD represents both cases in terms of the same relation type, i.e. `aux`. There are significant differences between English and Italian which might justify a strategy according to which auxiliaries and words used for introducing infinitival complements are treated in the same way. In English, open clausal complements are always introduced by the particle ‘to’, whereas in Italian different prepositions can introduce them (i.e. ‘a’, ‘di’, ‘da’) which are selected by the governing head. From this, it follows that the SD representation of the element introducing infinitival complements and modifiers in terms of `aux` might not be appropriate as far as Italian is concerned and it would be preferable to have a specific relation for dealing with introducers of infinitival complements (like `complm` in the case of finite clausal complements).

Another interesting and more complex example can be found for what concerns the partitioning of the space of sentential complements. MIDT distinguishes between `mod`(ifiers) on the one hand and subcategorised `arg`(uments) on the other hand: note that whereas `arg` is restricted to clausal complements subcategorized for by the governing head, the `mod` relation covers different types of modifiers (nominal, adjectival, verbal, adverbial, etc.). By contrast, SD resorts to distinct relation types depending on whether the clause is a subcategorized complement or a modifier (see e.g. `ccomp` vs. `advcl`), or whether the governor is a verb, a noun or an adjective (see e.g. `xcomp` vs. `infmod`), or whether the clause is headed by a finite or non-finite verb (see e.g. `ccomp` vs. `xcompl`). Starting from MIDT, the finer-grained SD distinctions within the class of clausal complements can be recovered by combining

dependency information with morpho-syntactic one (e.g. the mood of the verbal head or the morpho-syntactic category of the governing head).

Consider now head selection criteria. Due to their sharing similar design principles, MIDT and SD agree on the treatment of tricky cases such as the determiner–noun relation within nominal groups, the preposition–noun relation within prepositional phrases as well as the auxiliary-main verb relation in complex verbal groups. In both cases, head selection follows from a combination of syntactic and semantic criteria: i.e. whereas in the determiner–noun and auxiliary-verb constructions the head role is assigned to the semantic head (noun/verb), in preposition–noun constructions the head role is played by the element which is subcategorized for by the governing head, i.e. the preposition which is the syntactic head but can also be seen as a kind of role marker. In this area, the only but not negligible difference is concerned with subordinate clauses whose head in SD is assumed to be the verb, rather than the introducing element (whether a preposition or a subordinating conjunction) as in MIDT: in this case, the MIDT to SD conversion requires restructuring of the parse tree.

As already observed in the previous case study, further important differences are concerned with the treatment of coordination and punctuation. In both MIDT and SD schemes, coordinate constructions are considered as asymmetric structures with a main difference: while in MIDT both the conjunction and conjuncts starting from the second one are linked to the immediately preceding conjunct, in SD the conjunction(s) and the subsequent conjunct(s) are all linked to the first one. For what concerns punctuation, MIDT has its own linguistically-motivated strategy to deal with it, whereas SD does not appear to provide explicit and detailed annotation guidelines in this respect.

Focusing instead on dependency types which belong to an annotation scheme without a counterpart in the other, we find relation types which are not explicitly encoded in the MIDT source annotation, like the `ref` dependency linking the relative word introducing the relative clause and its antecedent, or the `xsubj` relation which in spite of its being part of the original TUT and ISST [2] resources have been omitted from the most recent and CoNLL-compliant versions, which represent the starting point of MIDT: in both cases, the "one head per dependent" constraint of the CoNLL representation format is violated. From this it follows that the SD conversion won't include these dependency types. Other SD relations which were part of the original TUT and ISST–TANL treebanks but were neutralised in MIDT are concerned with semantically-oriented distinctions which turned out to be problematic to be reliably identified in parsing in spite of their being explicitly encoded in both source annotation schemes [30]. This is the case of the indirect object relation (`iobj`) or of temporal modifiers (`tmod`).

Among the MIDT relation types which do not have a corresponding relation in SD, we find instead those typically representing Italian-specific peculiarities. This is the case of the `clit`(ic) dependency, linking clitic pronouns to the verbal head they refer to. In MIDT, whenever appropriate clitic pronouns are assigned a label that reflects their grammatical function (e.g. "dobj" or "iobj"): this is the case of reflexive constructions (*Maria si lava* lit. 'Maria her washes' meaning that 'Maria washes herself') or of complements overtly realized as clitic pronouns (*Giovanni*

mi ha dato un libro lit. 'Giovanni to-me has given a book' meaning that 'Giovanni gave me a book'). With pronominal verbs, in which the clitic can be seen as part of the verbal inflection, a specific dependency relation (`clit`) is resorted to link the clitic pronoun to the verbal head: for instance, in a sentence like *la sedia si è rotta* lit. 'the chair it is broken' meaning that 'the chair broke', the dependency linking the clitic *si* to the verbal head is `clit`. Other Italian-specific issues are concerned with the representation of sentential complements, as already pointed out above. The conversion process followed to generate ISDT starting from MIDT, is based on the results of the comparative analysis of the annotation schemes summarised above, and can be seen as organized in two different steps:

1. The first step is aimed at generating an enriched version of the MIDT resource, henceforth referred to as MIDT++, including SD-relevant distinctions originally neutralized in MIDT. During this step, relevant distinctions were recovered from the native resources which were lost in the conversion to MIDT, because of choices previously made in the design of the MIDT annotation scheme or simply because the harmonization of annotation styles was difficult without manual revision. This was the case, for instance, of: the annotation of indirect objects, present in both resources, and represented in MIDT as a generic *comp* relation; appositions, annotated only in the TUT resource, which where recovered from ISST–TANL with some heuristics and manual annotation; temporal modifiers, annotated in both resources, which were recovered only for the cases foreseen in the SD annotation scheme. The resulting augmented resource, MIDT++, is used here as a "bridge" towards SD;
2. The second step is in charge of converting the MIDT++ annotation in terms of the Stanford Dependencies as described in [25]. Starting from the results of the comparative analysis summarised above conversion patterns were defined, which can be grouped into two main classes according to whether they refer to individual dependencies (case A) or they involve dependency subtrees due to head reassignment (case B). Case A is handled in terms of *structure-preserving mapping rules* involving dependency retyping without restructuring of the tree: we distinguish here *1:1 mapping rules*, requiring dependency retyping only (e.g. MIDT `prep` > SD `pobj`, or MIDT `subj` > SD `nsubj`), and *1:n mapping rules*, requiring finer-grained dependency retyping (e.g. MIDT `mod` > SD `abbrev` | `amod` | `appos` | `nn` | `nnp` | `npadvmod` | `num` | `number` | `partmod` | `poss` | `preconj` | `predet` | `purplcl` | `quantmod` | `tmod`). Case B is instead treated with *tree restructuring mapping rules*, involving both head reassignment and dependency retyping: also in this case, we distinguish *1 : 1 versus 1 : n dependency mapping rules*.

To give the reader the flavor of how the abstract patterns described above have been translated into MIDT_to_SD conversion rules, consider the sentence *Giovanni ha dichiarato ai giudici di aver pagato i terroristi*, lit. 'Giovanni told to-the judges to have paid the terrorists', 'Giovanni told the judges that he has paid the terrorists', whose MIDT and SD representation is reported in Fig. 1a and 1b respectively. It can be noticed that in the conversion of the MIDT `arg` relation referring to a clausal

complement both head restructuring and dependency retyping have been performed. In MIDT, clausal complements, either finite or non-finite clauses, are linked to the governing head (which can be a verb, a noun or an adjective) as `arg`(uments), with a main difference with respect to SD, i.e. that the head of the clausal complement is the word introducing it (be it a preposition or a subordinating conjunction) rather than the verb heading the clausal complement. Depending on whether the clausal complement is headed by a finite verb or not the target SD relation changes: given that in this case we are dealing with an infinitival clause, the appropriate SD relation is `xcomp`, as it can be seen in Fig. 1b.

The conversion from MIDT to SD is still ongoing: we are currently evaluating alternative SD representations of problematic syntactic annotation areas, such as sentential complementation. Simultaneously, we started testing the resulting ISDT resource for parsing, in particular for training the DeSR parser, as it was done in the previous case study. For these initial experiments on the ISDT resource we used a basic and fast variant of the DeSR parser, the one based on Multi-Layer Perceptron (MLP) without reverse revision. Similarly, no parser combination was attempted. In fact, the purpose of the experiment was not to optimize the parser for the new resource but to compare the relative performances of the same parser on different versions of the ISDT resource, with the final aim of assessing the impact of different annotation choices on the parsing results.

The different performance of the parser on the two converted datasets (TUT–ISDT and ISST–TANL–ISDT) is in line with what was observed in previous experiments with native resources and MIDT [10, 30]; also in this case, the composition of the

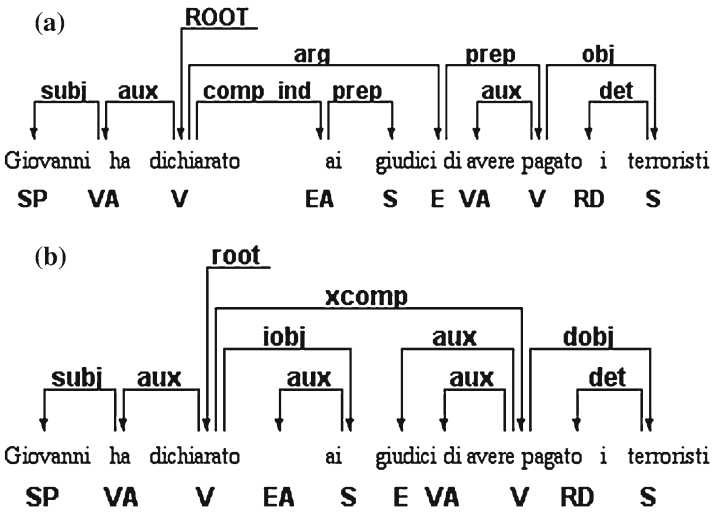

Fig. 1 MIDT versus SD annotation of the sentence *Giovanni ha dichiarato ai giudici di aver pagato i terroristi*, 'Giovanni told the judges that he has paid the terrorists' **a** MIDT representation. **b** SD representation

training and test corpora can be identified as a possible cause for such a difference, after the conversion. The preliminary results achieved using ISDT for training are encouraging, in line with what was obtained on the WSJ for English and reported in [31], where the best results in labeled attachment precision, achieved by a fast dependency parser (Nivre Eager feature Extract), is 81.7. For the time being, training with the larger combined resource does not seem to provide a substantial advantage, confirming results obtained with MIDT, despite the fact that in the conversion from MIDT to ISDT a substantial effort was spent to further harmonize the two resources.

6 Conclusion

In this paper, we addressed the challenge of combining and converting existing dependency-annotated resources with the final aim of costructing a bigger and standard-compliant treebank for the Italian language. The outcome of this effort is three-fold.

First, a methodology for harmonizing and merging annotation schemes belonging to the same family has been defined based on: a comparative analysis of the source and target annotation schemes, carried out with respect to different dimensions of variation, ranging from head selection criteria, dependency tagset granularity to defined annotation criteria; the analysis of the performance of state-of-the-art dependency parsers by using as training the source and the target treebanks; mapping of the source annotation scheme(s) onto a set of target set of (possibly underspecified) data categories. This methodology was tested in two different case studies aimed at (a) Combining existing resources (TUT and ISST–TANL) and (b) Converting the resource resulting from (a) into the Stanford Dependencies *de facto* annotation standard.

Second, Italian has now a bigger treebank, the *Merged Italian Dependency Treebank* (MIDT), which might be further extended if the other available treebanks will be involved in this harmonization and merging process. Italian is also going to have soon a new standard-compliant resource, i.e. the *Italian Stanford Dependency Treebank* (ISDT), resulting from the conversion of MIDT into the SD annotation scheme: we believe that this further conversion step will significantly improve the usability of the resource.

Third, but not least important, we defined an annotation scheme to be used as a "bridge" between different dependency annotation "dialects", i.e. dependency-based annotation schemes specialized for the same language, Italian: this is the MIDT scheme which presents itself as the lowest common ground between the native TUT and ISST–TANL annotation schemes. Within the second case study, we are also specializing the Stanford Dependency annotation scheme to deal with the peculiarities of the Italian language.

References

1. Bosco, C., Lombardo, V., Lesmo, L., Vassallo, D.: Building a treebank for Italian: a data-driven annotation schema. In: Proceedings of the 2nd Language Resources and Evaluation Conference (LREC'00), pp. 99–105. ELRA, Athens, Greece (2000)
2. Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M.T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F., Delmonte, R.: Building the Italian Syntactic-Semantic Treebank. In: Abeillé, A. (ed.) Building and Using Syntactically Annotated Corpora, pp. 189–210. Kluwer, Dordrecht (2003)
3. Tonelli, S., Delmonte, R., Bristot, A.: Enriching the Venice Italian Treebank with dependency and grammatical relations. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08), pp. 1920–1924. ELRA, Marrakech, Morocco (2008)
4. Buch-Kromann, M., Korzen, I., Müller, H.H.: Uncovering the 'lost' structure of translations with parallel treebanks. Spec. Issue Cph. Stud. Lang. **38**, 199–224 (2009)
5. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the EMNLP-CoNLL, pp. 915–932 (2007)
6. Montemagni, S., Simi, M.: The Italian dependency annotated corpus developed for the CoNLL-2007 shared task. Technical report, ILC-CNR (2007)
7. Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A.: Evalita'09 parsing task: comparing dependency parsers and treebanks. In: Proceedings of Evalita'09, Reggio Emilia, Italia (2009)
8. Dell'Orletta, F., Marchi, S., Montemagni, S., Venturi, G., Agnoloni, T., Francesconi, E.: Domain adaptation for dependency parsing at Evalita 2011. In: Working Notes of Evalita'11, Roma, Italia (2012)
9. Francesconi, E., Montemagni, S., Peters, W., Wyner, A. (eds.): Proceedings of the LREC Workshop on Semantic Processing of Legal Texts (SPLeT 2012). ELRA, Istanbul, Turkey (2012)
10. Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A., Lesmo, L., Attardi, G., Simi, M., Lavelli, A., Hall, J., Nilsson, J., Nivre, J.: Comparing the influence of different treebank annotations on dependency parsing. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10), pp. 1794–1801. ELRA, Valletta, Malta (2010)
11. Clegg, A.B., Shepherd, A.J.: Benchmarking natural-language parsers for biological applications using dependency graphs. BMC Bioinform. **8**(1), 17–24 (2007)
12. de Marneffe, M., Manning, C.: The Stanford typed dependencies representation. In: Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2008)
13. Leech, G., Barnett, R., Kahrel, P.: EAGLES recommendations for the syntactic annotation of corpora. Technical report, EAG-TCWG-SASG1.8 (1996)
14. Ide, N., Romary, L.: Representing linguistic corpora and their annotations. In: Proceedings of the 5th Language Resources and Evaluation Conference (LREC'06), pp. 225–228. ELRA, Genova, Italy (2006)
15. Ide, N., Suderman, K.: GrAF: A graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop (LAW'07), pp. 1–8. ACL, Prague, Czech Republic (2007)
16. Declerck, T.: SynAF: towards a standard for syntactic annotation. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08), pp. 229–232. ELRA, Marrakech, Morocco (2008)
17. Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., Wright, S.: ISOcat: remodelling metadata for language resources. IJMSO **4**(4), 261–276 (2009)

18. Bosco, C., Montemagni, S., Simi, M.: Converting Italian treebanks: towards an Italian Stanford dependency treebank. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse at ACL, pp. 61–69. ACL (2013)
19. Hudson, R.: Word Grammar. Basil Blackwell, Oxford (1984)
20. Lenci, A., Montemagni, S., Pirrelli, V., Soria, C.: A syntactic meta-scheme for corpus annotation and parsing evaluation. In: Proceedings of the 2nd Language Resources and Evaluation Conference (LREC'00), pp. 625–632. ELRA, Athens, Greece (2000)
21. Kübler, S., McDonald, R., Nivre, J.: Dependency Parsing. Morgan and Claypool, Oxford and New York (2009)
22. Cheung, J., Penn, G.: Topological field parsing of German. In: Proceedings of the ACL-IJCNLP'09, pp. 64–72. ACL, Suntec, Singapore (2009)
23. Ide, N., Bunt, H.: Anatomy of annotation schemes: mapping to graf. In: Proceedings of the 4th Linguistic Annotation Workshop (LAW IV'10), pp. 247–255. Stroudsburg (2010)
24. Hayashi, Y., Declerck, T., Narawa, C.: LAF/GrAF-grounded representation of dependency structures. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10), pp. 1990–1995. ELRA, Valletta, Malta (2010)
25. de Marneffe, M., Manning, C.: Stanford typed dependencies manual. Stanford University, Technical report, CA (2008)
26. Attardi, G.: Experiments with a multilanguage non-projective dependency parser. In: Proceedings of the CoNLL-X'06, pp. 166–170. New York City, New York (2006)
27. Attardi, G., Dell'Orletta, F.: Reverse revision and linear tree combination for dependency parsing. In: Proceedings of the NAACL HLT'09, pp. 261–264. Boulder, Colorado (2009)
28. Attardi, G., Dell'Orletta, F., Simi, M., Turian, J.: Accurate dependency parsing with a stacked multilayer perceptron. In: Proceedings of Evalita'09, Reggio Emilia, Italy (2009)
29. Bosco, C., Mazzei, A.: The Evalita dependency parsing task: from 2007 to 2011. In: Magnini, B., Cutugno, F., Falcone, M., Pianta, E. (eds.) Evaluation of Natural Language and Speech Tools for Italian, pp. 1–12. Roma, Italia (2012)
30. Bosco, C., Simi, M., Montemagni, S.: Harmonization and merging of two Italian dependency treebanks. In: Proceedings of the LREC Workshop on Language Resource Merging, pp. 23–30. ELRA, Istanbul, Turkey (2012)
31. Cer, D.M., de Marneffe, M.C., Jurafsky, D., Manning, C.D.: Parsing to Stanford dependencies: trade-offs between speed and accuracy. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10), pp. 1628–1632. ELRA, Valletta, Malta (2010)

Dependency Treebank Annotation and Null Elements: An Experiment with VIT

Rodolfo Delmonte

Abstract State of the art parsers are currently trained on converted versions of Penn Treebank into dependency representations, which however don't include null elements. This is done to facilitate structural learning and prevents the probabilistic engine to postulate the existence of deprecated null elements everywhere, see [19]. However it is a fact that in this way the semantics of the representation used and produced is inconsistent and will reduce dramatically its usefulness in real life applications, like Q/A and other semantically driven fields, by hampering the mapping of a complete logical form. What systems have come up with are "quasi"-logical forms or partial logical forms mapped directly from the surface representation in dependency structure. We show the most common problems derived from the conversion and then describe an algorithm that we have implemented to apply to our converted Italian Treebank, that can be used on any CoNLL-like treebank or representation to produce an almost complete semantically consistent dependency treebank.

Keywords Dependency representations · Treebanks

1 Introduction

In this chapter I shall present a symbolic rule-based algorithm that takes as input CoNLL-style dependency-based representations and populates them with all lexically unexpressed and implicit linguistic elements, excluding elliptical material. I have been working with two languages, Italian and English, but I assume that the algorithm can be applied to any language provided a sub-categorization computational lexicon—it could also be based on FrameNet, VerbNet, PropBank, WordNet—is available for the language. The algorithm also computes best semantic roles to associate to arguments and adjuncts, and provides antecedents for all types of controlled

R. Delmonte (✉)
Department of Language and Cultural Studies, Department of Computer Science, Ca' Foscari University of Venice, Venezia, Italy
e-mail: delmont@unive.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_2

empty subjects. It also makes use of a pronominal anaphora resolution algorithm, which however only gives a preference antecedent that requires manual checking.

We tested the algorithm on a fragment of VIT, the Venice Italian Treebank, which contains 500 sentences and 15,000 tokens and we ended up with an addition of over 600 new items fully co-indexed. Providing unexpressed and implicit linguistic items is a paramount process to enable semantic predicate argument representations to be produced automatically. This is not only an essential step for a complete linguistic resource such as a treebank, but also for any follow up, be it machine learning for grammar induction or for application oriented scenarios, which require—as for instance Question Answering—fully implemented predicate-argument structures.

Missing implicit or unexpressed linguistic elements can be of three types:

- unexpressed SUBJects of untensed clauses (including infinitivals, participials, gerundives be they computable as arguments or as adjuncts of a given predicate);
- unexpressed SUBJects of tensed clauses, this being highly language specific, whereas Italian freely allows to leave unexpressed the subject of tensed clause, English will only allow it in imperatives and coordinate clauses;
- traces, or empty linguistic items in what are called "long distance dependency" constructions, like relative clauses and interrogative clauses.

For every new added empty linguistic item, the algorithm looks for the antecedent on which the item will be dependent—this can be local for most of the cases, but it can also be external to the sentence where the empty item has been added. In the latter case, the antecedent can be definite and fully referential, or it can be indefinite or better generic, thus non referential. This may apply to impersonal pronouns and to untensed clauses with generic reference, which I will revise below. In the case of Italian, SUBJects of tensed classed will search for the antecedent in a previous stretch of discourse with an anaphoric binding algorithm which is not the topic of this paper, but see [18].

We have been referring to CoNLL style column representation used in the CoNLL shared tasks series of conferences, which are a conversion of Penn Treebank [27] portions by means of Lund's University tool. In fact, the conversion contains many mistakes, which badly ruin the semantic import of the output. In this section we shall comment on some examples before presenting our algorithm.

In the following section we will be showing treatment also in other treebanks besides PT and VIT: we will refer to TUT, by the University of Turin [26] (http://www.di.unito.it/~tutreeb/), ISST–TANL by ILC-CNR and the University of Pisa [32], AnCORA—by the University of Barcelona [28], and the Portuguese Treebank, made available at CoNLL-X by Afonso et al. [1], that we call CPT.

2 "Ne" in Relative Clauses

We can foresee three types of syntactic control for relative and interrogative clauses:

- **Direct Control**: whenever the pronoun or surface complementizer are bound to a core argument, a subject, an object, an indirect object or an oblique, and the dependency is with the verb of the governed clause;
- **Indirect Control**: whenever the pronoun or surface complementizer constitutes a modifier or specifier of a complement of the verb of the relative clause, for instance with predicative complements in copulative clauses, and the dependency is with the complement and not with the verb;
- **Double Control**: or Pied Piping, whenever the pronoun usually contained in a nominal or prepositional constituent, modifies a local nominal head. In turn, the whole structure modifies a complement of the relative clause. So two dependencies should be annotated: first the one of the relative pronoun with a local nominal head, and then the ensuing dependency with some complement in the relative clause structure. In the following sections we will be talking about these different types of dependencies and show how treebanks have annotated them.

2.1 "Che" as Beginner of a Relative Clause

In shallow or surface dependency treebank, relative pronouns are only visible if lexically expressed. So the case of implicit relative pronoun signaled by "*che*" complementizer does not exist—but it does in all deep dependency treebanks as we will show below. What is usually done is the transformation of "*che*" itself into a relative pronoun like "*chi*", "*cui*" or "*quale*" and others. We will discuss other cases below. However, even though this is what all shallow treebanks do, the treatment of "*che*" is not uniform. We look at first into ISST-TANL:

(1) *produrre individui che sanno fare cose che essi non potranno mai nemmeno immaginare…*

21	produrre	produrre	V	V	mod=f	5	sub
22	individui	individuo	S	S	num=p|gen=m	21	obj
23	che	che	P	PR	num=n|gen=n	24	subj
24	sanno	sapere	V	V	num=p|per=3|mod=i|ten=p	22	mod_rel
25	fare	fare	V	V	mod=f	24	arg
26	cose	cosa	S	S	num=p|gen=f	25	obj
27	che	che	P	PR	num=n|gen=n	33	obj
28	essi	essi	P	PE	num=p|per=3|gen=m	33	subj
29	non	non	B	BN	_	33	neg
30	potranno	potere	V	VM	num=p|per=3|mod=i|ten=f	33	modal
31	mai	mai	B	B	_	33	mod_temp
32	nemmeno	nemmeno	B	B	_	33	mod
33	immaginare	immaginare	V	V	mod=f	26	mod_rel

As it can be easily noticed, "*che*" is only treated as functional dependent on the verb of Relative Clause. As functional dependent it shouldn't be associated to any semantic function: on the contrary it is marked as SUBJect of the verb "*pervadere*" and "*conservare*". These verbs are then linked as dependents to the head noun governing the relative pronoun. The pronoun in this case coincides with "*che*", thus introducing a specialization for verb relations in addition to linking to other verbs, or to subordinating/coordinating conjunctions. TUT treatment is identical to ISST-TANL:

(2) *Nelle società di Tirana che hanno truffato…c'è…*

1	Nelle	IN PREP	PREP	MONO	15	RMOD
2	Nelle	IL ART	ART	DEF|F|PL	1	ARG
3	societa'	SOCIETÀ	NOUN	noun common|f|allval	2	ARG
4	di	DI PREP	PREP	MONO	3	RMOD
5	Tirana	TIRANA	NOUN	noun proper|f|sing|££city	4	ARG
6	che	CHE	PRON	PRON relat|allval|allval|lsubj+lobj	8	SUBJ
7	hanno	AVERE	VERB	VERB aux|ind|pres|trans|3|pl	8	AUX+TENSE
8	truffato	TRUFFARE	VERB	VERB main|participle|past|trans|sing|m	3	RMOD+RELCL
…						
14	c'	CI PRON	PRON	loc|allval|allval|loc|clitic	15	RMOD
15	e'	ESSERE	VERB	VERB main|ind|pres|intrans|3|sing	0	TOP

In this example, the prepositional phrase headed by "*nelle*" is linked to the main verb "*essere*" in 15, and the noun "*società*" governing the relative clause is linked to "*nelle*"; "*truffare*" the verb of the relative clause is linked to "*società*", but the information as to what grammatical function this head noun plays in the relative clause is indicated in the "*che*".

The recovery of grammatical relations for "*truffare*"—and the same will apply to previous cases of ISST–TANL—will go through a process of restructuring of the argument subject "*società*" with "*che*" that works as functional head but does not have any explicit link with it. This is different from what happens in VIT:

(3) *emergere di una crescente concorrenza che si è progressivamente spostata…*

13 emergere emergere n(noun) sn num=s|gen=m 12 pobj com
14 di di pd(preposition_di) spd - 13 mod nil
15 una uno art(article) sn num=s|gen=f 17 sn ind
16 crescente crescente ag(adjective) sa num=s|per=fm 17 mod nil
17 concorrenza concorrenza n(noun) sn num=s|gen=f 14 pobj com
18 che che rel(relative) f2 - 17 subj-theme_aff nil
19 si si clit(clitic_pronoun) ibar per=3|gen=m|num=sp 22 ibar acc
20 è essere ause(auxiliary_essere_tensed) ibar punt 22 ibar aux
21 progressivamente progressivamente avv(adverb) ibar [] 22 adjv mn
22 spostata spostare vppin(verb_intrans_past_participle) ibar punt 18 ibar refl_in/posit

In this case we see that "*che*" is bound to its noun head "*concorrenza*", which has a certain role in the sentence to which it belongs, headed by "*emergere*" through preposition "*di*"; then *che* is the intermediary between the head noun and verb of the relative "*spostare*" which is linked to it. The role of "*che*" is played, in the case of

a deep representation, by the empty category. This prevents "*che*" itself to carry the information of both grammatical function and semantic role, as happens in previous treebanks.

Here below is what the Portuguese treebank has for relatives, where we see that the head noun of the relative is linked and has roles in the main clause, the relative pronoun carried the roles it has in the relative clause and is linked to its verb, the verb of the relative clause is linked to the head noun:

(4) *milhões que o Ministério_do_Planeamento_e_Administração_do_Território já gasta…*

7	milhões	milhão	n	n	F|P	4	P<	
8	que	que	pron	pron-indp	<rel>|M|P	12	ACC	
9	o	o	art	art	<artd>|M|S	10	>N	
10	Ministério_do_Planeamento_e_Administração_do_Território							
	Ministério_do_Planeamento_e_Administração_do_Território					prop	prop M|S	12
	SUBJ							
11	já	já	adv	adv	_	12	ADVL	
12	gasta	gastar	v	v-fin	PR|3S|IND	7	N<	

This is identical to what AnCORA, the Catalan treebank has done.

(5) *el treball que es desplaça…*

26	el	el	d	da	num=s|gen=m	27	ESPEC
27	treball	treball	n	nc	num=s|gen=m	24	CD
28	que	que	p	pr	num=n|gen=c	30	SUJ
29	es	es	p	p0	_	30	PASS
30	desplaça	desplaçar	v	vm	num=s|per=3|mod=i|ten=p	27	SF

Now, the difference in treatment is clear and can be summarized below. We have come up with two different approaches to the problem of treating "*che*"/ "*que*"/ "*that*" relative pronoun—which by the way is identical to what happens with *che* complementizer of sentential complements:

- linked to the governing Noun head (VIT)
  - the governed verb of the relative clause linked to *che*
- linked to the governed verb in the relative clause (all other treebanks)
  - the governed verb being an auxiliary if present (TUT, PTB)
  - the governed verb being the lexical semantic verb (all other treebanks)

In the following section we will propose a treatment of relative pronouns, which is based on surface dependency structure but introduces the concept of chain. To substantiate our proposal we now show the output of deep treebanks, like the one proposed by PARC–XEROX and organized on the basis of LFG theoretical framework. This treebank is based on the same set of newspaper articles from "Wall Street Journal" of PTB and in particular it contains all articles belonging to section 23. We will indicate only relevant dependency nodes to highlight differences from previous treebanks.

(6) *But not much money was spent on the shows\, either\, a situation that encouraged cheap-to-make talk and game shows\, while discouraging expensive-to-produce dramas.*

```
pron_rel(encourage~12, pro~16)
subj(encourage~12, pro~16)
topic_rel(encourage~12, pro~16)
case(pro~16, nom)
num(pro~16, sg)
pers(pro~16, 3)
pron_form(pro~16, that)
pron_type(pro~16, relative)
adjunct(situation~7, encourage~12)
adjunct_type(encourage~12, relative)
```

"*That*" is treated as complementizer and is linked to PRO relative. PRO is marked with NOMINATIVE and is linked to "*encourage*", the verb of the relative. "*Situation*" is linked to "*encourage*", and is linked to PRO. This is partially coincident with what VIT has done and partially with TUT and ISST–TANL. In the deep treebank, it is the implicit pronoun PRO, which plays the role of intermediary between the complementizer "*that*" and the verb of the relative. The relative clause as a whole is then linked as dependent to the head noun "*situation*".

From a computational point of view, a chain allows more easily a recovery of all relations needed in case of further processing of dependency structures for semantic purposes. The chain goes from the relative pronoun to its noun head binder, however all relevant information is already encoded in the ADJUNCT additional entry, where the relative pronoun has already undergone head substitution with its binder antecedent. The index carried by the null pron_rel is the same as that of the verb of the relative, in this way partly resembling the linking of the verb to relative head noun. The complementizer can in this case simply be done away with in the semantics, as would happen with the case of sentential complements, being a functional head with no semantic content.

If we look at PTB original constituency structure, we see that relative pronouns are embedded in the NP of the head noun they depend on; then the relative pronoun and its preposition if existent, and following head noun in the case of "*whose*", are all included in a SBAR structure, that is the relative clause, that they are beginners of. Finally, the index associated to the relative pronoun is then "landed" in a position around the verb of the relative clause, either just before or after in case of argument relation, or after the expressed arguments in case of adjunct relation as shown below:

(7) *three levels on which to treat the subject…*

```
(NP (NP three levels)
    (SBAR (WHPP-1 on
              (WHNP which))
          (S (NP-SBJ *)
             (VP to
                (VP treat
                   (NP the subject)
                   (PP-LOC *T*-1))))))
```

(8) *I don't know what to do…*

```
(S (NP-SBJ-2 I)
   (VP do n't
      (VP know
         (SBAR (WHNP-1 what)
            (S (NP-SBJ *-2)
               (VP to
                  (VP do
                     (NP *T*-1)
                     (PP-CLR with
                           (NP this sentence)))))))))
```

In some cases as the following one, the trace lands deeper below, inside an infinitival, complement of the main verb of the relative clause:

(9) *The following prompts allow you to specify how you want the printed output to look…*

```
(S (NP-SBJ The following prompts)
      (VP allow
         (S (NP-SBJ you)
            (VP to
               (VP specify
                  (SBAR (WHADVP-1 how)
                     (S (NP-SBJ you)
                        (VP want
                           (S (NP-SBJ the printed output)
                              (VP to
                                 (VP look
                                    (ADVP-MNR *T*-1)))))))))))))
```

Penn Treebank also signals with traces passive constructions so that in case of a passive relative clause the number of traces is doubled.

There are no principled reasons for not using a chain-like description of the relative clause structure, from what is contained in this annotation. If embedding is used to detect dependency, then the relative pronoun should always be dependent on the head noun it is governed by. The presence of the trace in the following clause should then be used to make the verb of the relative clause dependent on the relative pronoun. Romance languages have a much wider inventory of relative pronouns than German ones, in particular Italian has certainly the most extended one, and we will discuss them in the section below.

2.2 Lexical Relative Pronouns

Lexical pronouns have a different status from "*che*" complementizer at least in as far as they would contain internally enough information to an independent semantic specification. In fact, relative pronouns can also be subdivided by the traditional categorization of "analytic" ("*il quale*", etc.) versus "synthetic" ("*che*", "*cui*") pronouns: this subdivision, however, is irrelevant to the discussion about dependency

structure. We will look into "*cui*" and "*quale*" preceded or not by preposition. From the structures below, we see that the same technique is being used for linking relative pronouns and their prepositions: dependency links are established as before, between the verb of the relative clause which is made dependent upon the nominal head of the relative pronoun; then the preposition is made dependent on the verb of the relative clause, and the relative pronoun on the preposition. In all the examples below, recovering the binder and noun antecedent of the relative pronoun requires at least a search in two steps as will be explained below. We will start by looking at excerpts from TUT deep:

(10) *nei luoghi abituali in cui di TV si parla…*

31 nei (IN PREP MONO) [30;PREP-RMOD-LOC+IN]
31.1 nei (IL ART DEF M PL) [31;PREP-ARG]
32 luoghi (LUOGO NOUN COMMON M PL) [31.1;DET+DEF-ARG]
33 abituali (ABITUALE ADJ QUALIF ALLVAL PL) [32;ADJC+QUALIF-RMOD]
34 in (IN PREP MONO) [39;PREP-RMOD-LOC+IN]
35 cui (CUI PRON RELAT LIOBJ+OBL) [34;PREP-ARG]
36 di (DI PREP MONO) [39;VERB-INDCOMPL-THEME]
37 Tv (|Tv| NOUN PROPER) [36;PREP-ARG]
38 si (SI PRON REFL-IMPERS ALLVAL ALLVAL 3 LSUBJ+LOBJ+LIOBJ CLITIC) [39;VERB-SUBJ/VERB-SUBJ+IMPERS]
39 parla (PARLARE VERB MAIN IND PRES INTRANS 3 SING) [32;VERB-RMOD+RELCL]

2.2.1 Pied Piping Relative Pronouns

In this section we present relative pronouns headed by a preposition which in turn are embedded in another prepositional phrase that is then governed by the nominal head of the relative:

(11) *è il coronamento del dialogo di cui oggi si vedono i risultati…*

8 e` (ESSERE VERB MAIN IND PRES INTRANS 3 SING) [0;TOP-VERB]
9 il (IL ART DEF M SING) [8;VERB-PREDCOMPL+SUBJ]
10 coronamento (CORONAMENTO NOUN COMMON M SING CORONARE TRANS) [9;DET+DEF-ARG]
11 del (DI PREP MONO) [10;NOUN-OBJ]
11.1 del (IL ART DEF M SING) [11;PREP-ARG]
12 dialogo (DIALOGO NOUN COMMON M SING) [11.1;DET+DEF-ARG]
13 di (DI PREP MONO) [17;VISITOR]
14 cui (CUI PRON RELAT LIOBJ+OBL) [13;PREP-ARG]
15 oggi (OGGI ADV TIME) [17;ADVB-RMOD-TIME]
16 si (SI PRON REFL-IMPERS ALLVAL ALLVAL 3 LSUBJ+LOBJ+LIOBJ CLITIC) [17;VERB-SUBJ/VERB-SUBJ+IMPERS]
17 vedono (VEDERE VERB MAIN IND PRES TRANS 3 PL) [12;VERB-RMOD+RELCL]
18 i (IL ART DEF M PL) [17;VERB-OBJ]
19 risultati (RISULTATO NOUN COMMON M PL RISULTARE INTRANS) [18;DET+DEF-ARG]
19.10 t [13f] (DI PREP MONO) [19;NOUN-SUBJ]

This seems the only case in which a trace is inserted to allow for the genitive "*di cui*" to be linked appropriately as complement of "*risultati*". However here again in

order to get the antecedent of "*cui*", which is the nominal head "*dialogo*", one has to search the verb.

(12) *lo studente di Ancona scomparso e del cui caso si era occupata…*

29 lo (IL ART DEF M SING) [23;APPOSITION]
30 studente (STUDENTE NOUN COMMON M SING) [29;DET+DEF-ARG]
31 di (DI PREP MONO) [30;PREP-RMOD-LOC+ORIGIN]
32 Ancona (ANCONA NOUN PROPER F £CITY) [31;PREP-ARG]
33 scomparso (SCOMPARIRE VERB MAIN PARTICIPLE PAST INTRANS SING M) [30;VERB-RMOD+RELCL+REDUC]
40 e (E CONJ COORD COORD) [33;COORD+BASE]
41 del (DI PREP MONO) [46;VERB-INDCOMPL-THEME]
41.1 del (IL ART DEF M SING) [41;PREP-ARG]
42 cui (CUI PRON RELAT LIOBJ+OBL) [43;PRON-RMOD]
43 caso (CASO NOUN COMMON M SING) [41.1;DET+DEF-ARG]
44 si (SI PRON REFL-IMPERS ALLVAL ALLVAL 3 LSUBJ+LOBJ+LIOBJ CLITIC) [46;EMPTYCOMPL]
45 era (ESSERE VERB AUX IND IMPERF INTRANS 3 SING) [46;AUX+TENSE]
46 occupata (OCCUPARE VERB MAIN PARTICIPLE PAST TRANS SING F) [40;COORD2ND+BASE]

Recovering the antecedent in this case requires climbing the coordinate structure, then from the reduced relative "*scomparso*" finding the nominal head "*studente*". However this seems to be identical to the previous example where "*risultati*" was lacking its complement: "*caso*" perhaps should have been followed by a trace that identified its complement clearly, in this way the genitive would have been made explicit.

The same remarks can be made if we look at ISST–TANL, where the relation intervening between the relative oblique pronoun and its nominal head binder is not available and must be recovered indirectly from the verb:

(13) *Forma in cui si presenta in natura…*

1	Forma	forma	S	S	num=s|gen=f	0	ROOT
2	in	in	E	E	_	5	comp
3	cui	cui	P	PR	num=n|gen=n	2	prep
4	si	si	P	PC	num=n|gen=n	5	clit
5	presenta	presentare	V	V	num=s|per=3|mod=i|ten=p	1	mod_rel
6	in	in	E	E	_	5	comp
7	natura	natura	S	S	num=s|gen=f	6	prep
8	.	.	F	FS	_	1	punc

2.2.2 Lexical Relative Pronouns in Other Treebanks

ISST–TANL and TUT encode relations in the same way in which PTB and other treebanks do, as shown below:

(14) *and should honor local convictions about which tasks most need doing…*

27	and	_	CC	_	_	17	COORD
28	should	_	MD	_	_	27	CONJ
29	honor	_	VB	_	_	28	VC
30	local	_	JJ	_	_	31	NMOD
31	convictions	_	NNS	_	_	29	OBJ
32	about	_	IN	_	_	31	NMOD
33	which	_	WDT	_	_	34	NMOD
34	tasks	_	NNS	_	_	36	SBJ
35	most	_	RBS	_	_	36	ADV
36	need	_	VBP	_	_	32	PMOD
37	doing	_	VBG	_	_	36	OBJ

In all these examples, the preposition is linked to the verb of the relative clause, "*which*" is linked to the preposition, and the verb of the relative is in turn linked to the head noun of the relative pronoun. To show the difference intervening between "*which*" and "whose"—that is somewhat comparable to "*cui*"— we will now present some examples with the genitive relative pronoun "*whose*"—that is always accompanied by at least a noun or a noun with modifiers—which resembles in some way the sequence (ART) "*cui*", but without preceding articles.

(15) *Sony, whose innovative premium-priced products are …*

2	Sony	_	NNP	_	_	18	SBJ
3	,	_	,	_	_	2	P
4	whose	_	WP$	_	_	8	NMOD
5	innovative	_	JJ	_	_	8	NMOD
6	,	_	,	_	_	8	P
7	premium-priced	_	JJ	_	_	8	NMOD
8	products	_	NNS	_	_	9	SBJ
9	are	_	VBP	_	_	2	NMOD

(16) *Kollmorgen, whose agreement to be acquired for $25 a share by Vernitron collapsed …*

3	Kollmorgen	_	NNP	_	_	2	OBJ
4	,	_	,	_	_	3	P
5	whose	_	WP$	_	_	6	NMOD
6	agreement	_	NN	_	_	17	SBJ
7	to	_	TO	_	_	6	NMOD
8	be	_	VB	_	_	7	IM
9	acquired	_	VBN	_	_	8	VC
10	for	_	IN	_	_	9	ADV
11	$	_	$	_	_	10	PMOD
12	25	_	CD	_	_	11	NMOD
13	a	_	DT	_	_	14	NMOD
14	share	_	NN	_	_	11	ADV
15	by	_	IN	_	_	9	LGS
16	Vernitron	_	NNP	_	_	15	PMOD
17	collapsed	_	VBD	_	_	3	NMOD

As can be noticed, "*whose*" requires a totally different treatment from "*which*": it is linked to the head noun it modifies—it specifies its semantic content—and this noun

is then linked to the verb of the relative clause. The verb of the relative is then linked to the head noun but this noun does not modify the verb, in fact it does not have any relation with it being a modifier of one of the arguments of the relative clause. I indicate here below in brackets the position of "*whose*" and of its lexical substitute:

innovative products [*of Sony*] *are*/*Sony* [*whose*] *innovative products are*
agreement [*of Kollmorgen*] *collapsed*/*Kollmorgen* [*whose*] *agreement collapsed*

For this reason, I don't see why the verb of the relative should be linked to the head noun of the relative pronoun, rather than directly to the relative pronoun, and the latter in turn linked to the head noun.

In the case of "*which*", the relations are different: relative pronoun, "*which*" is governed by the preposition, which is heading an adjunct or argument of the verb of the relative itself. Very much the same would happen with simple relative pronouns, which are arguments of the verb of the relative. So eventually, the treatment of "*whose*"/"*cui*" seems inadequate in particular in view of its mapping onto a semantic predicate-argument structure. To see in more depth the ways in which the mapping of oblique/genitive relative pronouns may take place we look into PARC–700 relevant portions to check how the LFG has decided to encode it. We look at few different examples and we see that the treatment is definitely organized on the basis of the presence of a NULL element, pro. What is important to stress here is the fact that "*whose*" expresses a possessive genitive relation with its local head that it modifies, and that this relation is represented by an abstract "*pro*" linked to "*whose*" and from there in a chain with the head noun, and then linked to the verb of the relative: "*respect*", "*sing*", "*be*", "*determine*", "*keep*".

(17) *But Mr. Davis, whose views are widely respected by money managers, says he expects no 1987-style crash.*

adjunct(Mr. Davis∼1, respect∼18)
adjunct(respect∼18, widely∼24)
adjunct_type(respect∼18, relative)
obl_ag(respect∼18, manager∼19)
pron_rel(respect∼18, pro∼22)
subj(respect∼18, view∼20)
topic_rel(respect∼18, view∼20)
mod(manager∼19, money∼28)
pcase(manager∼19, by)
poss(view∼20, pro∼22)
pron_form(pro∼22, whose)
pron_type(pro∼22, relative)

(18) *One of Italy's favorite shows, "Fantastico", a tepid variety show, is so popular that viewers clamored to buy a chocolate product, "Cacao Fantastico", whose praises were sung each week by dancing showgirls—even though the product didn't exist.*

adjunct_type(Cacao Fantastico~61, parenthetical)
poss(praise~62, pro~64)
pron_form(pro~64, whose)
pron_rel(sing~54, pro~64)
pron_type(pro~64, relative)
subj(sing~54, praise~62)
obl_ag(sing~54, showgirl~43)
adjunct_type(sing~54, relative)
topic_rel(sing~54, praise~62)
adjunct(product~92, Cacao Fantastico~61)

For these reasons, the role of "*cui*" in particular has been given a lot of attention in the deep version of VIT, that we comment here below.

2.3 "Cui" in VIT

There at least four different typologies of structure accompanying "*cui*" oblique relative pronoun, that we have found in VIT:

1. argument/adjunct of relative verb
   – it directly modifies the main verb of the relative clause
2. adjunct modifier of argument of relative verb
3. – it modifies an argument of the verb of relative clause
4. adjunct modifier of a noun
5. adjunct modifier of the internal nominal head

All of the following examples show the variety of cases in which "*cui*" can act as an adjunct but also as an argument with different semantic roles:

2.3.1 Argument/Adjunct of Relative Verb

All of the following examples show the variety of cases in which CUI can act as an adjunct but also as an argument with different semantic roles:

(19) *dell'ambiente socio-economico in cui sono inserite …*

38 dell di partd(preposition_di_plus_article) spd num=s|per=fm 49 mod det
38.1 l il art sn num=s|per=fm 49 det def
39 ambiente ambiente n(noun) sn num=s|gen=m 38 pobj com
40 socio_economico socio_economico ag(adjective) sa num=s 39 mod nil
41 in in p(preposition) sp - 39 adj nil
42 cui cui relob(relative_oblique) sn [] 41 binder rel_obl
43 sono essere ause(auxiliary_essere_tensed) ibar punt 44 ibar aux
44 inserite inserire vppt(verb_trans_past_participle) ibar punt 39 ibar refl_in/into_hole

44.11 prep_relob in_ambiente prep_relob(prepositional_rel_oblique) sp num=s |gen=m ant=41_42 bindee com

(20) *nella norma in cui si stabilisce …*

7 nella in part(preposition_plus_article) sp num=s|gen=f 6 obl det
7.1 la il art sn num=s|gen=f 6 det def
8 norma norma n(noun) sn num=s|gen=f 7 pobj com
9 in in p(preposition) sp - 8 adj nil
10 cui cui relob(relative_oblique) sn [] 9 binder rel_obl
11 si si clit(clitic_pronoun) ibar per=3|gen=m|num=sp 12 ibar nom
12 stabilisce stabilire vt(verb_trans_tensed) ibar punt 8 ibar refl/exten
12.10 pro si pro(little_pro) sn per=3|gen=m|num=sp 11 s_impers-agent nom
12.11 prep_relob in_norma prep_relob(prepositional_rel_oblique) sp num=s|gen =f ant=9_10 bindee com

(21) *cose più importanti di cui occuparmi …*

2 cose cosa n(noun) sn num=p|gen=f 1 ncomp com
3 più più in(intensifier) sa [] 4 sa q
4 importanti importante ag(adjective) sa num=p|per=fm 2 mod nil
5 di di pd(preposition_di) spd - 2 adj nil
6 cui cui relob(relative_oblique) sn [] 5 binder rel_obl
7 occuparmi occupare vcl(verb_with_enclitic) sv2 punt 2 adj tr
7.11 prep_relob di_cosa prep_relob(prepositional_rel_oblique) sp num=p|gen=f ant=5_6 bindee com

Here "*di cui*" is *argument* of the main verb of the relative clause, "*occuparmi*".

2.3.2 Adjunct Modifier of Predicate Argument of Relative Verb

The samples in this subsection are all referred to the special case of copulative constructions as relative clauses, in which the oblique relative is a modifier of the predicate, usually an adjective.

(22) *il costo al quale la sostituzione è possibile e la misura in cui è fattibile …*

31 il il art(article) sn num=s|gen=m 32 sn def
32 costo costo n(noun) sn num=s|gen=m 39 ncomp com
33 al a part(preposition_plus_article) sp num=s|gen=m 32 mod det
33.1 l il art sn num=s|per=fm 32 det def
34 quale quale rel(relative) f2 num=s|gen=m 33 binder nil
35 la il art(article) sn num=s|gen=f 36 sn def
36 sostituzione sostituzione n(noun) sn num=s|gen=f 37 subj-tema_bound com
37 è essere vc(verb_copulative) ibar punt 34 ibar cop/esistenza
38 possibile possibile ag(adjective) sa num=s|per=fm 37 acomp nil
39 e e cong(conjunction) coord [] 30 coord sum
40 la il art(article) sn num=s|gen=f 41 sn def

41 misura misura n(noun) sn num=s|gen=f 39 ncomp com
42 in in p(preposition) sp - 41 adj nil
43 cui cui relob(relative_oblique) sn [] 42 binder rel_obl
44 è essere vc(verb_copulative) ibar punt 41 ibar cop/esistenza
44.10 pro sostituzione pro(little_pro) sn num=s|per=3 ant=36 s_impl-tema_bound nil
45 fattibile fattibile ag(adjective) sa num=s|per=fm 44 acomp nil
45.11 prep_relob in_misura prep_relob(prepositional_rel_oblique) sp num=s|gen =f ant=42_43 bindee com

Even though "*la_misura_in_cui*" may sometimes be used as adverbial locution, in this case it is just the SUBJect of "*be*" and consequently "*cui*" is head of relative clause that modifies "*fattibile*"—"*in …misura*". The same applies to the example below, where the relative pronoun is a modifier of "*responsabile*".

(23) *sulle due branche operative, di cui pure è nominalmente responsabile …*

36 sulle su part(preposition_plus_article) sp num=p|gen=f 32 pcomp det
36.1 le il art sn num=p|gen=f 32 det def
37 due due num(numeral) sn [] 38 sn card
38 branche branca n(noun) sn num=p|gen=f 36 pobj com
39 operative operativo ag(adjective) sa num=p|gen=f 38 mod nil
40 , , punt(sentence_internal) sn punt 38 sn nil
41 di di pd(preposition_di) spd—38 adj nil
42 cui cui relob(relative_oblique) sn [] 41 binder rel_obl
43 pure pure cong(conjunction) f2 [] 38 cong sum
44 è essere vc(verb_copulative) ibar punt 38 ibar cop/esistenza
45 nominalmente nominalmente avv(adverb) savv [] 46 adjm mn
46 responsabile responsabile ag(adjective) sa num=s|per=fm 44 acomp nil
46.11 prep_relob di_branca prep_relob(prepositional_rel_oblique) sp num=p|gen =f ant=41_42 bindee com

2.3.3 Adjunct Modifier of Noun Argument of Relative Verb

In this subsection, the oblique relative is a modifier of a nominal predicate, "*bisogno*" and further on "*candidato*".

(24) *fondi di cui abbiamo bisogno …*

21 fondi fondo n(noun) sn num=p|gen=m 19 obj com
22 di di pd(preposition_di) spd - 21 adj nil
23 cui cui relob(relative_oblique) sn [] 22 binder rel_obl
24 abbiamo avere vc(verb_copulative) ibar nil 21 ibar cop/stato
24.10 pro pro pro(little_pro) sn nil ant=7 s_impl-esperiente impL1p
25 bisogno bisogno n(noun) sn num=s|gen=m 24 ncomp com
25.11 prep_relob di_fondo prep_relob(prepositional_rel_oblique) sp num=p|gen =m ant=22_23 bindee com

(25) *commissione esteri alla cui presidenza è candidato …*

8 commissione commissione n(noun) sn num=s|gen=f 6 pobj com
9 esteri estero ag(adjective) sa num=p|gen=m 8 mod nil
10 alla a part(preposition_plus_article) sp num=s|gen=f 8 adj det
10.1 la il art sn num=s|gen=f 8 det def
11 cui cui relob(relative_oblique) sp [] 10 sp rel_obl
12 presidenza presidenza n(noun) sn num=s|gen=f 10 pobj com
13 è essere vc(verb_copulative) ibar punt 8 ibar cop/esistenza
14 candidato candidato n(noun) sn num=s|gen=m 13 ncomp com
14.11 prep_relob alla_commissione prep_relob(prepositional_rel_oblique) sp num =s|gen=m ant=10_11 bindee com

2.3.4 Adjunct Modifier of Embedded Argument of Relative Verb

This example shows a case of oblique relative which is a modifier of an argument embedded in an infinitival complement of a process verb "*continuare*", very much like the example we saw from PTB in the section above:

(26) *una strategia di cui tutti i ministri interessati continuano a sottolineare la collegialità …*

0 Una uno art(article) sn num=s|gen=f 1 sn ind
1 strategia strategia n(noun) sn num=s|gen=f 13 sn com
2 di di pd(preposition_di) spd - 1 adj nil
3 cui cui relob(relative_oblique) sn [] 2 binder rel_obl
4 tutti tutto qc(quantifier_collective) sq num=p|gen=m 6 sq nil
5 i il art(article) sn num=p|gen=m 6 sn def
6 ministri ministro n(noun) sn num=p|gen=m 8 subj-exper com
7 interessati interessato ppas(past_participle_absolute) sa num=p|gen=m 6 mod nil
8 continuano continuare vt(verb_trans_tensed) 3 ibar - ibar raisn/process
9 a a pt(verbal_participle) sv2 - 10 sv2 nil
10 sottolineare sottolineare vit(verb_trans_infinitive) sv2 punt 8 vcomp tr
10.10 pPro pPro pPro(big_pro) sn nil ant='6' s_impl-causer ministro
11 la il art(article) sn num=s|gen=f 13 sn def
12 < < par(parenthetical) sn - 13 sn nil
13 collegialità collegialità n(noun) sn num=f 10 obj invar
13.11 prep_relob di_strategia prep_relob(prepositional_rel_oblique) sp num=s |gen=f ant=1_2 bindee com

In this example, the relative pronoun modifies "*collegialità*", and the semantics should compose the following pseudo-structure:

una strategia [*di cui*] *tutti i ministri interessati continuano a sottolineare la collegialità* [t] → *la collegialità* [*della strategia*]

2.3.5 Adjunct Modifier of an Ellipsed Nominal Head

Not all cases of relative pronouns are connected to a fully lexicalized relative clause: there are cases in which the clause is unexpressed—as would happen with reduced relatives—but also ellipsed as shown in the following examples:

(27) *nomi di rilievo, tra cui l'ex ministro della difesa …*

12 nomi nome n(noun) sn num=p|gen=m 11 obj-theme_unaff com
13 di di pd(preposition_di) spd - 12 mod nil
14 rilievo rilievo n(noun) sn num=s|gen=m 13 pobj com
15 , , punt(sentence_internal) sn punt 12 sn nil
16 tra tra p(preposition) sp - 12 adj nil
17 cui cui relob(relative_oblique) sn [] 16 binder rel_obl
18 l il art(article) sn num=s|gen=m 20 sn def
19 ex ex ag(adjective) sa num=f|gen=m 20 mod invar
20 ministro ministro n(noun) sn num=s|gen=m 17 subj com
20.11 prep_relob tra_nome prep_relob(prepositional_rel_oblique) sp num=p|gen =m ant=16_17 bindee com
21 della di partd(preposition_di_plus_article) spd num=s|gen=f 20 mod det
21.1 la il art sn num=s|gen=f 20 det def
22 difesa difesa n(noun) sn num=s|gen=f 21 pobj com

The peculiarity of this structure is the fact that it is a fragment, which however has a main nominal head: to complete the semantics it could be enriched by the presence of a "dummy *be*" verb, or perhaps a dummy "*there_be*", so that the head noun "*ministro*" becomes subject of predication. The oblique relative modifies directly the subject nominal "*ministro*" or indirectly, in case of presence of dummy "*be*", through the predication:

→ *l'ex ministro … E' tra i nomi*

The same applies to the example below:

(28) *collaboratori… tra cui il capo della polizia …*

29 collaboratori collaboratore n(noun) sn num=p|gen=m 28 pobj com
…
37 tra tra p(preposition) sp - 29 adj nil
38 cui cui relob(relative_oblique) sn [] 37 binder rel_obl
39 il il art(article) sn num=s|gen=m 40 sn def
40 capo capo n(noun) sn num=s|gen=m 38 sn com
40.11 prep_relob tra_collaboratori prep_relob(prepositional_rel_oblique) sp num=p|gen=m ant=37_38 bindee com
41 della di partd(preposition_di_plus_article) spd num=s|gen=f 40 mod det
41.1 la il art sn num=s|gen=f 40 det def
42 polizia polizia n(noun) sn num=s|gen=f 41 pobj com

2.3.6 Adjunct of the Subject/Object Nominal Head of the Relative

Eventually, we also find cases in which the relative "*cui*" modifies the SUBJect head noun it depends on, as is the case in the example below:

(29) *Non sarà presente, invece, l'uomo … la cui posizione è stata stralciata …*

0 Non non neg(negation) ir_infl - 1 neg nil
1 sarà essere vcir(verb_copulative_mood_irrealis) cl(main) punt - ir_infl cop/esistenza
2 presente presente ag(adjective) sa num=s|per=fm 1 acomp nil
3 , , punt(sentence_internal) compc punt 1 compc nil
4 invece invece congf(conjunction_sentential) compc [] 1 cong av
5 , , punt(sentence_internal) compc punt 1 compc nil
6 l il art(article) sn num=s|gen=m 7 sn def
7 uomo uomo n(noun) sn num=m 1 s_top-tema_bound invar
…
19 la il art(article) sn num=s|gen=f 21 sn def
20 cui cui relob(relative_oblique) sn [] 7 sn rel_obl
21 posizione posizione n(noun) sn num=s|gen=f 24 subj-theme_unaff com
21.11 prep_relob di_uomo prep_relob(prepositional_rel_oblique) sp num=s|gen=m ant=7 bindee com
22 è essere ause(auxiliary_essere_tensed) ibar punt 24 ibar aux
23 stata essere ausep(auxiliary_essere_past_participle) ibar punt 24 ibar aux
24 stralciata stralciare vppt(verb_trans_past_participle) ibar punt 21 ibar tr/possess

In this structure the oblique is only active locally even though the main verb would occur in the following portion of the sentence, it does not contribute to the following relative clause structure, neither as argument nor as adjunct nor as modifiers of some argument.

2.4 Oblique Relative in Online Parsers

We already saw in the previous section the treatment of "*whose*" in PTB. As an experiment I tried out a sentence which contained a pied piped oblique genitive in English, with both CONNEXOR and STANFORD parsers to see the relations they encode in the output see [2]. However none of the output is able to show differences in treatment from previous examples.

(31) *John, in whose house the accident took place, is leaving home.*

```
1   John        john        @OBJ %NH N NOM SG
2   ,           ,
3   in          in          @ADVL %EH PREP
4   whose       who         attr:>5     @A> %>N <Rel> PRON WH GEN
5   house       house       @<P %NH N NOM SG
6   the         the         det:>7      @DN> %>N DET
7   accident    accident    subj:>8     @SUBJ %NH N NOM SG
8   took        take        pcomp:>3            @+FMAINV %VA V PAST
9   place       place       obj:>8      @OBJ %NH N NOM SG
10  ,           ,
11  is          be          v-ch:>12 @+FAUXV %AUX V PRES SG3
12  leaving     leave       @-FMAINV %VA ING
13  home        home        goa:>12  @ADVL %EH N NOM SG
14  now         now         tmp:>12  @ADVL %EH ADV
15  .           .
16  <s>         <s>
```

And this is the output of Stanford parser:

Typed dependencies	**Typed dependencies, collapsed**
nsubj(leaving-10, John-1)	nsubj(leaving-10, John-1)
prep(John-1, in-2)	poss(house-4, whose-3)
poss(house-4, whose-3)	dobj(took-7, house-4)
dobj(took-7, house-4)	det(accident-6, the-5)
det(accident-6, the-5)	nsubj(took-7, accident-6)
nsubj(took-7, accident-6)	prepc_in(John-1, took-7)
pcomp(in-2, took-7)	dobj(took-7, place-8)
dobj(took-7, place-8)	aux(leaving-10, is-9)
aux(leaving-10, is-9)	root(ROOT-0, leaving-10)
root(ROOT-0, leaving-10)	dobj(leaving-10, home-11)
dobj(leaving-10, home-11)	advmod(leaving-10, now-12)
advmod(leaving-10, now-12)	

What is missing, then here, is the information that "*the house*" belongs to John, and the role of *whose* is left unexplained.

2.5 Questions

Questions are hard to parse for statistical parsers, given their sparsity in available treebanks. ISST–TANL, as TUT does, encodes the relation intervening between the interrogative pronoun and the verb of the relative directly by linking it to the verb.

(32) *Perché avete ucciso altri albanesi?*

```
1 Perche' (|perche'| ADV INTERR) [3;ADVB+INTERR-RMOD]
2 avete (AVERE VERB AUX IND PRES TRANS 2 PL) [3;AUX+TENSE]
3 ucciso (UCCIDERE VERB MAIN PARTICIPLE PAST TRANS SING M) [0;TOP-VERB]
3.10 t [] (DEITT-T PRON PERS M PL 2) [3;VERB-SUBJ]
4 altri (ALTRO ADJ DEITT M PL) [3;VERB-OBJ]
5 albanesi (ALBANESE NOUN COMMON ALLVAL PL) [4;DET+DEF-ARG]
6 ? (#\? PUNCT) [3;END]
```

(33) *Vediamo cosa si può fare.*

6 Vediamo (VEDERE VERB MAIN IND PRES TRANS 1 PL) [0;TOP-VERB]
6.10 t [] (DEITT-T PRON PERS ALLVAL PL 1) [6;VERB-SUBJ]
7 cosa (COSA PRON INTERR ALLVAL SING LSUBJ+LOBJ) [9;VISITOR]
8 si (SI PRON REFL-IMPERS ALLVAL ALLVAL 3 LSUBJ+LOBJ+LIOBJ CLITIC) [9;VERB-SUBJ/VERB-SUBJ+IMPERS]
9 puo` (POTERE VERB MOD IND PRES INTRANS 3 SING) [6;VERB-OBJ]
10 fare (FARE VERB MAIN INFINITE PRES TRANS) [9;VERB+MODAL-INDCOMPL]
10.10 t [8f] (SI PRON REFL-IMPERS ALLVAL ALLVAL 3 LSUBJ+LOBJ+LIOBJ CLITIC) [10;VERB-SUBJ]
10.11 t [7f] (COSA PRON PERS INTERR) [10;VERB-OBJ]

The following case is very interesting: we have a purpose infinitival clause governed by "*vedere*" which has as complement an indirect interrogative clause headed by a pronoun which has an anaphoric link with an antecedent ("*banche*") that is placed in the main clause. This is correctly marked with a trace, as if it were a syntactically governed relation.

(34) *Il privato cittadino che fa il giro delle banche per vedere in quale guadagnerebbe…*

4 il (IL ART DEF M SING) [3;VERB-OBJ]
5 privato (PRIVATO ADJ QUALIF M SING) [6;ADJC+QUALIF-RMOD]
6 cittadino (CITTADINO NOUN COMMON M SING) [4;DET+DEF-ARG]
7 che (CHE PRON RELAT ALLVAL ALLVAL LSUBJ+LOBJ) [8;VERB-SUBJ]
8 fa (FARE VERB MAIN IND PRES TRANS 3 SING) [6;VERB-RMOD+RELCL]
9 il (IL ART DEF M SING) [8;VERB-OBJ]
10 giro (GIRO NOUN COMMON M SING) [9;DET+DEF-ARG]
11 delle (DI PREP MONO) [10;PREP-RMOD]
11.1 delle (IL ART DEF F PL) [11;PREP-ARG]
12 banche (BANCA NOUN COMMON F PL) [11.1;DET+DEF-ARG]
13 per (PER PREP MONO) [8;PREP-RMOD-REASONCAUSE]
14 vedere (VEDERE VERB MAIN INFINITE PRES TRANS) [13;PREP-ARG]
14.10 t [7p] (CHE PRON RELAT ALLVAL ALLVAL LSUBJ+LOBJ) [14;VERB-SUBJ]
15 in (IN PREP MONO) [17;PREP-RMOD-LOC+METAPH]
16 quale (QUALE ADJ INTERR ALLVAL SING LSUBJ+LOBJ) [15;PREP-ARG]
16.10 t [12f] (BANCA NOUN COMMON F PL) [16;DET+INTERR-ARG]
17 guadagnerebbe (GUADAGNARE VERB MAIN CONDIZ PRES TRANS 3 SING) [14;VERB-OBJ]
17.10 t [14.10p] (CHE PRON RELAT ALLVAL ALLVAL LSUBJ+LOBJ) [17;VERB-SUBJ]

In the following sentence the SUBJect relation is reverted and the predicative complement is positionally rather than semantically determined: the interrogative pronoun that precedes the main verb for structural constraints is wrongly computed as SUBJ of the predication. The noun phrase "*i politici*" which should be the legitimate SUBJect is computed as a predication. This is a case of subject inversion, which is very common in Italian, and not only in this language—very difficult to detect in general.

(35) *Chi sono i politici …*

1 Chi (CHI PRON INTERR ALLVAL ALLVAL LSUBJ+LOBJ) [2;VERB-SUBJ]
2 sono (ESSERE VERB MAIN IND PRES INTRANS 3 PL) [0;TOP-VERB]
3 i (IL ART DEF M PL) [2;VERB-PREDCOMPL+SUBJ]
4 politici (POLITICO NOUN COMMON M PL) [3;DET+DEF-ARG]

In another portion of TUT we see that the relations are correctly annotated:

(36) *Qual'è il pericolo di contrarre l'infezione nel corso di un rapporto occasionale.*

5 Qual (QUALE PRON INTERR ALLVAL SING 3 LSUBJ+LOBJ+OBL) [6;VERB-PREDCOMPL+SUBJ]
6 è (ESSERE VERB MAIN IND PRES INTRANS 3 SING) [0;TOP-VERB]
7 il (IL ART DEF M SING) [6;VERB-SUBJ]
8 pericolo (PERICOLO NOUN COMMON M SING) [7;DET+DEF-ARG]
9 di (DI PREP MONO) [8;VERB+INF-RMOD-SITDESCR]
10 contrarre (CONTRARRE VERB MAIN INFINITE PRES TRANS) [9;PREP-ARG]
10.10 t [] (GENERIC-T PRON PERS ALLVAL ALLVAL ALLVAL) [10;VERB-SUBJ]
11 l' (IL ART DEF F SING) [10;VERB-OBJ]
12 infezione (INFEZIONE NOUN COMMON F SING) [11;DET+DEF-ARG]
13 nel (NEL_CORSO_DI PREP POLI LOCUTION) [10;PREP-RMOD-TIME]
14 corso (NEL_CORSO_DI PREP POLI LOCUTION) [13;CONTIN+LOCUT]
15 di (NEL_CORSO_DI PREP POLI LOCUTION) [14;CONTIN+LOCUT]
16 un (UN ART INDEF M SING) [13;PREP-ARG]
17 rapporto (RAPPORTO NOUN COMMON M SING) [16;DET+INDEF-ARG]
18 occasionale (OCCASIONALE ADJ QUALIF ALLVAL SING) [17;ADJC+QUALIF-RMOD]

And now we will look into VIT:

(37) *Cosa risponde loro?*

0 Cosa cosa int(interrogative_pronoun) fint num=s 3 fint int
1 risponde rispondere vt(verb_trans_tensed) cl(main) punt - ibar trans/dir_speech
1.10 pro pro pro(little_pro) sn num=s|per=3 ant=sent_00195/10 s_impl-agente nil
1.11 rel_pro cosa rel_pro(relative_pronoun) bindee num=s|gen=m ant=0 obj-info com
2 loro egli pron(pronoun) sn num=p|per=fm 1 iobj pers
3 ? ? puntint(punctuation_non_declarative) fint nil 1 fint puntint

If we look at CPT and ADT we can see that their treatment interrogative clauses is identical to ISST-TANL and TUT, in that the interrogative pronoun is directly linked to the following verb, and in case it is headed by a preposition, it is linked to the preposition which in turn is linked to the following verb.

3 Creation of Null Elements

We mapped constituency-based VIT onto dependency structure and came up with a structure lacking all NEs see [17]. Eventually VIT looked very similar to the output of current state-of-the-art statistical treebank parsers trained on PTB. So we imagined that we could create a script or algorithm to try and produce all null elements and try to coindex them automatically, in line with what other researchers have done for Chinese, for example which has similar problems—left-dislocation and unexpressed subject, in particular, in addition Italian has also right dislocation and clitics see also [25]. We selected 500 complex sentences from VIT, with average sentence length of 30 tokens, total tokens 15,000. However, before starting work on the algorithm, we realized soon that some ambiguity had to be solved manually or else our automatic procedure would never be able to come to a reasonable solution see [29, 31]. I am referring to a manual classification of "*si*" (pro)clitic which is a cause

of difficulty even for the most skilled annotators. When we worked at the construction of the annotation manual for ISST national project for the Italian treebank, we came up together with colleagues from Pisa unit to the following fine-grained classification for "*si*":

- "*si*" passivizing, diat=middle, syn_form=pers, reflex= passive
- impersonal, with "*si*", diat=active, syn_form=si_impers
- intransitive pronominal, with "*si*", diat=middle, syn_form=pers, reflex= ipron
- reflexive, with "*si*", diat=middle, syn_form=pers, reflex=rifl
- reflexive apparent, diat=middle, syn_form=pers, reflex=rifl_app
- reflexive apparent con "*ci_si*", diat=middle, syn_form=si_impers, reflex=rifl_app
- reflexive con "*ci_si*", diat=middle, syn_form=si_impers, reflex=rifl

We then eventually agreed on what is computationally relevant, that is the distinction between "impersonal *si*", "reflexive *si*", and "expletive or pleonastic *si*". These three cases have however to be distinguished manually. Differentiating "middle" cases would be beneficial for Semantic Role assignment because it is always the case that the deep object has been raised to become the subject. However, introducing this additional feature would have made the classification impossible to complete in a short period of time.

After completing this work we went back to the algorithm, which is organized in different steps as follows.

The first step has been the annotation of all missing subject of tensed clauses, what is usually called the little_pro instance of empty subject pronoun. This is clearly a preliminary step in that it is then mandatory to complete the argument structure of each clause before dealing with "untensed" clauses, that is infinitivals, participials and gerundives. This process is itself organized as the addition of a null element with the same index of the governing verb, which was then diversified by the association of an additional number, 11. Then we wanted to add features coming from the antecedent and from the verb; the real problem then was finding the antecedent: to that aim we recovered our anaphora resolution algorithm and adapted it to the task. But then we discovered that only a percentage of all little_pros required an anaphora resolution algorithm, 31.4 %. The remaining cases had local antecedents of different types or were simply expletive subjects, as shown in Table 1.

Second Step was a procedure to classify "*si*" clitic pronoun, which is based on the nature of the governing verb. We use a syntactic classification associated to our verb lexicon of Italian made up of 17,000 entries, which encompasses 237 different

Table 1 Little_pros in portion of VIT

Type of real.	Freq. occurr.
Discourse	70
subj_expl	47
subj_impers	38
subj_impl+ant	65
Total little_pro	223

categories. In addition we look for presence of an expressed Subject and Object. From the experiment on 500 sentences we evaluated 92 % accuracy of the algorithm.

Third step is the recovery of so-called *wh*-traces in relative and interrogative clauses, otherwise treated as long-distance dependencies in LFG. We found 286 cases of null elements of this type, which we formalize as follows:

Case 1 Implicit Argument/Adjunct with relative pronoun as local antecedent
(49) *concorrenza che si è progressivamente spostata/competition which has increasingly moved*

17 concorrenza concorrenza noun sn num=s|gen=f 14 pobj com
18 che che relative f2 - 17 binder nil
19 si si clit ibar per=3|gen=f|num=sp 22 ibar acc
20 è essere ause ibar punt 22 ibar aux
21 progressivamente progressivamente avv ibar [] 22 adjv mn
22 spostata spostare vppin ibar punt 18 ibar refl_in/posit
22.11 rel_pro concorrenza rel_pro bindee num=s|per=3|md='L'|ts='K' ant=17 subj-theme_aff nil

Fourth step is the recovery of the unexpressed subject of dislocated and fronted infinitivals, which work as subject clauses. Fifth step is the assignment of expletive little_pro to subjects of lexically determined weather and impersonal verbs. Sixth step is the assignment of NE subject to tenseless clauses, which is formalized as big_pPro. We found 139 occurrences of this type of null element, which is represented with the antecedent index and also the head, as follows:

Case 2 Implicit Subject with local antecedent
(50) *ad aumentare l'efficienza/to increase the efficiency*

22 ad ad pt sv2 - 23 sv2 nil
23 aumentare aumentare vit sv2 punt 21 adj tr/exten
23.11 pPro pPro big_pro sn nil ant='10' s_impl-agent infrastruttura
24 l_ il article sn num=s|gen=f 25 sn def
25 efficienza efficienza noun sn num=s|gen=f 23 obj com

The examples below illustrate the output of the manual and automatic annotation: as far as verbs are concerned, we introduced both a fine-grained syntactic category and a semantic class taken from our subcategorized lexicon; for arguments and adjuncts we added semantic roles by a bottom up procedure that chose the best frame according to available information. Here are some excerpts of the new updated VIT with null subject elements classified:

Case 3 Impersonal Subject
(51) *quando si arriva/when one arrives*

18 quando quando cosu fs [] 20 fs temp
19 si si clit ibar per=3|gen=m|num=sp 20 ibar nom
20 arriva arrivare vin ibar punt 30 ibar unac/posit

20.11 pro si little_pro sn per=3|gen=m|num=sp 19 s_impers-theme_unaff nom

Case 4 Implicit Subject with local antecedent
(52) *e dipenderà/and it will depend*

11 e e cong fc [] 8 fc sum
12 dipenderà dipendere virin ir_infl punt 11 ir_infl unac/exten
12.11 pro pro little_pro sn num=s|per=3|md='U'|ts='K' ant=1 s_impl-theme_unaff nil

Case 5 Expletive Subject
(53) *ed è in questa quota che/and it is in this share that*

12 ed ed cong fc [] 4 fc sum
13 è essere vc ibar punt 12 ibar cop/existence
13.11 pro pro little_pro nil num=s|per=3|md='L'|ts='K' 17 s_expl nil
14 in in preposition sp - 13 pcomp nil
15 questa questo dim sa num=s|gen=f 16 mod nil
16 quota quota noun sn num=s|gen=f 14 pobj com
17 che che complementizer fac - 16 fac nil

Case 6 Expletive Subject with *si* antecedent
(54) *si tratta di/it deals with*

0 Si si clit ibar - 1 ibar nil
1 tratta trattare vin cl(main) punt - ibar refl/exten
1.11 pro si little_pro nil num=s|gen=m ant=0 s_expl com
2 del di partd spd num=s|gen=m 1 obl det

Case 7 Implicit Subject with relative pronoun antecedent
(55) *Berlusconi che è industriale/Berlusconi who is industrialist*

19 Berlusconi Berlusconi nh sn propr 15 s_top-experiencer hum
20 che che rel f2 - 19 binder nil
21 è essere vc ibar punt 23 ibar cop/existence
21.11 pro pro little_pro sn num=s|per=3|md='L'|ts='K' ant=19 s_impl-tema_bound nil
22 industriale industriale noun sn num=s 21 ncomp com

Case 8 Implicit Subject with Discourse antecedent
(56) *annaspa/it fumbles*

2 annaspa annaspare vin ibar punt 0 ibar unerg/exten
2.11 pro sisde little_pro sn punt ant=sent_00132/6 s_impl-theme_aff intr

As can be seen, we have six different notations associated with little_pro, which can be bound to impersonal "*si*", an expletive "*si*" or an extraposed sentential subject, a local antecedent, a relative pronoun as antecedent and finally a discourse level

antecedent where the nominal head is reported. In all other cases, morphological features are associated coming either from the verb or from the antecedent itself.

Overall we added 617 new fully annotated null elements. Then, we used this dataset as gold data to check the working of the algorithm: we ran the algorithm on the raw version of the dataset and matched the result with the gold augmented version of the dataset of the 500 sentences: we found 43 mistakes (that is 0.7 % error rate), most of which (32, that is 0.5 %) was a wrong antecedent for discourse bound little_pros. The fragment is now freely downloadable here, http://project.cgm.unive.it/?page_id=200.

4 Conclusion

As it appears, there are differences between the treebanks considered in this chapter. However, it is important to remind that the main difference lies in the decision to produce a deep versus a shallow structure see [10]. The deep structure may well encode dependency relations with the auxiliary help of null elements. I would say that major differences are in fact only minor problems that can be easily mended by an ad hoc script. What is not easy to produce is the presence of null Elements, which require a lot of additional computation and of manual checking.

The main difference is in the treatment of relative pronouns and relative clauses. In particular, we think some amendment is badly needed in the way "*cui*" and other pied piping constructions have been treated. These structures need to be represented differently from relative clauses headed by relative pronouns acting as direct argument/adjunct of the relative verb. Of course this is something that can be done at best by inserting some empty element. However in some cases, there is a need to check dependencies, which are not directly to the verb but to an argument/adjunct of the verb of the relative.

Luckily, these structures seem to be fairly uncommon. So eventually the net advantage in modifying a parser or some automatic procedure for the treebank annotation, is very small.

Other questions regard minor items, as said above, and they can be interpreted in terms of overall treebank consistency/coherence, and/or its strict/loose adherence to a linguistic theory. Treebank conversion tools made available for the CoNLL international challenge have determined a "*de facto*" standard in the way in which dependency relations are encoded. And this is obviously reflected in the fact that Penn Treebank has become the "*de facto*" standard of all syntactic treebanks. But it is clear that mapping constituency to dependency is not always easy and may require difficult decisions to be taken. Uniformity in the mapping encoded in a script is not always easy to guarantee, as we saw above. Also decisions as to what constitutes a HEAD in dependency terms is not an easy decision in some cases, even though the theoretical background of linguistic theories should be helpful if properly used. In particular, if functional heads are treated as dependents they should always be treated as such; the same applies to the opposite case. However, this might become

ambiguous in case there is a need to represent an implicit category like PRO for unexpressed relative pronouns.

Eventually what is needed is Semantic Transparency. In other words, annotations in treebanks should be as much as possible transparent to semantic mapping procedures if they are to be of any use at all. For this reason we are convinced of the following:

- Minor categories and functional heads should always be treated as dependent, or if needed, be part of a chain with a semantic head; this is particularly true for the case in which negation is linked to the auxiliary rather than the lexical verb.
- Preserving the original orthography is not a major issue of a dependency treebank; multi-words should be treated as one unit if that is semantically justified; amalgams should be decomposed if needed for semantic opportunity—enclitics constitute arguments that will undergo anaphoric processes, but incorporated articles don't need to be assigned a separate index.
- Positing the existence of an abstract category like COORD, which may serve for semantic purposes might be allowed even if it is linked to punctuation.

References

1. Afonso, S., Eckhard, B., Renato H., Diana S. : Floresta sintá(c)tica: a treebank for Portuguese. In: Rodríguez, M.G., Araujo, C.P. (eds.) Proceedings of LREC 2002, pp. 1698–1703. ELRA, Spain (2002)
2. Attardi, G.: Experiments with a multilanguage non-projective dependency parser. In: Proceedings of the Tenth Conference on Natural Language Learning, New York (2006)
3. Bikel, D.M.: Intricacies of Collins' parsing model. Comput. Linguist. **30**(4), 479–511 (2003)
4. Black, E., Abney, S., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B., Strzalkowski, T.: A procedure for quantitatively comparing the syntactic coverage of English grammars. In: Proceedings of the DARPA Speech and Natural Language Workshop, pp. 306–311 (1991)
5. Brants, T.: TnT: a statistical part-of-speech tagger. In: ANLP 2000. Seattle (2000)
6. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Comput. Linguist. **21**, 543–565 (1995)
7. Carroll, J., Briscoe, T., Sanfilippo, A.: Parser evaluation: a survey and a new proposal. In: Proceedings of the [First] International Conference on Language Resources and Evaluation, pp. 447–454 (1998)
8. Collins, Michael, : A new statistical parser based on bigram lexical dependencies. In: Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pp. 184–191 (1996)
9. Corazza, A., Lavelli, A., Satta, G., Zanoli, R.: Analyzing an Italian treebank with state-of-the-art statistical parsers. In: Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT-2004), pp. 39–50. Tübingen, Germany (2004)
10. Delmonte, R., Bristot, A., Tonelli, S.: VIT—Venice Italian Treebank: syntactic and quantitative features. In: De Smedt, K., Hajic, J. Kübler, S. (eds.) Proceedings Sixth International Workshop on Treebanks and Linguistic Theories. Nealt Pnealt Proceedings Series, vol. 1, pp. 43–54 (2007)
11. Delmonte, R., Luminita, C., Ciprian, B. : Elementary trees for syntactic and statistical disambiguation. In: Proceedings TAG+5, pp. 237–240. Paris (2000)
12. Delmonte, R.: From shallow parsing to functional structure. In: Atti del Workshop AI*IA—"Elaborazione del Linguaggio e Riconoscimento del Parlato", pp. 8–19. IRST, Trento (1999)

13. Delmonte, R.: How to annotate linguistic information in FILES and SCAT. In: Atti del Workshop "La Treebank Sintattico-Semantica dell'Italiano di SI-TAL", pp. 75–84. Bari (2001)
14. Delmonte, R.: Strutture Sintattiche dall'Analisi Computazionale di Corpora di Italiano. In: Anna Cardinaletti(a cura di), Intorno all'Italiano Contemporaneo, pp. 187–220. Franco Angeli, Milano (2004)
15. Delmonte, R., Dolci, R.: Parsing Italian with a context-free recognizer. Annali di Ca' Foscari **XXVIII**(1–2), 123–161 (1989)
16. Delmonte, R.: Shallow Parsing and Functional Structure in Italian Corpora, pp. 113–119. LREC, Atene (2000)
17. Delmonte, R.: Treebanking in VIT: from phrase structure to dependency representation. In: Nirenburg, Sergei (ed.) Language Engineering for Lesser-Studied Languages, pp. 51–80. IOS Press, The Netherlands (2009)
18. Delmonte, R.: Computational Linguistic Text Processing—Lexicon Grammar Parsing and Anaphora Resolution. Nova Science Publishers, New York (2009)
19. Gaizauskas R.: Investigations into the grammar underlying the Penn treebank II. Technical Report CS-95-25, Department of Computer Science, University of Sheffield (1995)
20. Harper, M.P., Helzerman, R.A.: Extensions to constraint dependency parsing for spoken language processing. Comput. Speech Lang. **9**, 187–234 (1995)
21. Hellwig, P.: Dependency unification grammar. In: Proceedings COLING-86, pp. 195–198 (1986)
22. Hudson, R.: Word Grammar. Blackwell, London (1984)
23. Hudson, R.: English Word Grammar. Blackwell, London (1990)
24. Jackendoff, R.: X-Bar Syntax. The MIT Press, Cambridge (1977)
25. Jaervinen, T., Tapanainen, P.: Towards an implementable dependency grammar. In: Kahane, S., Polguère, A. (eds.) Proceedings of the Workshop on Processing of Dependency-Based Grammars, pp. 1–10 (1998)
26. Lesmo, L., Lombardo, V., Bosco, C.: Treebank development: the TUT approach. In: Proceedings of ICON 2002. Mumbai (2002)
27. Marcus, M., et al.: Building a large annotated corpus of English: the Penn Treebank. Comput. Linguist. **19**(2), 313–330 (1993)
28. Martí, M.A., Taulé, M., Márquez, L., Bertran, M.: Ancora: A Multilingual and Multilevel Annotated Corpus in http://clic.ub.edu/ancora/publications/ (2007)
29. Maruyama, H.: Structural disambiguation with constraint propagation. In: Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL), pp. 31–38. Pittsburgh (1990)
30. Mel'cuk, I.: Dependency Syntax: Theory and Practice. State University of New York Press, New York (1988)
31. Menzel, W., Schroeder, I.: Decision procedures for dependency parsing using graded constraints. In: Kahane, S., Polguère, A. (eds.) Proceedings of the Workshop on Processing of Dependency-Based Grammars, pp. 78–87 (1998)
32. Montemagni, et al.: The Italian Syntactic-Semantic Treebank: Architecture, Annotation, Tools and Evaluation, pp. 18–27. LINC, ACL, Luxembourg (2000)

PartTUT: The Turin University Parallel Treebank

Manuela Sanguinetti and Cristina Bosco

Abstract In this paper, we introduce an ongoing project for the development of a parallel treebank for Italian, English and French. The treebank is annotated in a dependency format, namely the one designed in the Turin University Treebank (TUT), hence the choice to call such new resource Par(allel)TUT. The project aims at creating a resource which can be useful in particular for translation research. Therefore, beyond constantly enriching the treebank with new and heterogeneous data, so as to build a dynamic and balanced multilingual treebank, the current stage of the project is devoted to the design of a tool for the alignment of data, which takes into account syntactic knowledge as annotated in this kind of resource. The paper focuses in particular on the study of translational divergences and their implications for the development of the alignment tool. The paper provides an overview of the treebank, with its current content and the peculiarities of the annotation format, the description of the classes of translational divergences which could be encountered in the treebank, together with a proposal for their alignment.

Keywords Parallel treebanks · Translation

1 Introduction

Parallel corpora are currently considered as crucial resources for a variety of NLP tasks (most notably machine translation), and for research in the field of translation studies and contrastive linguistics. Their usefulness, as in the case of monolingual resources, increases when they are annotated and their annotations allow forms of alignment at various levels of linguistic knowledge. But the development of such resources raises several unsolved applicative and theoretical issues. First, the devel-

M. Sanguinetti (✉) · C. Bosco
Dipartimento di Informatica, Università di Torino,
Corso Svizzera 195, 10149 Torino, Italy
e-mail: msanguin@di.unito.it

C. Bosco
e-mail: bosco@di.unito.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_3

opment of treebanks is usually semi-automatically performed and is a very time-consuming and error-prone process. Second, all the possible levels of alignment of data, e.g. sentence, words or various syntactic components, can be in principle of some interest for the extraction of information relevant for translation and other tasks, but the development of alignment tools is currently limited to particular linguistic knowledge levels and annotation formats (statistical MT models have only recently begun to take advantage of higher-level linguistic structures). It is our belief, however, that linguistic insight can be of great help in linguistic applications, and alignment in particular, especially in identifying not only the exact matches, but also those cases in which there are partial or fuzzy correspondences due, for example, to the individual translator choices or to differences—which often occur in a systematic way—between language pairs. This is the reason why we decided to build such linguistic resource, on one hand, and to exploit linguistic information encoded in it to design a new alignment system for TUT parallel trees.

The assumptions on which the system design is based are two: that an efficient alignment requires linguistically informed approaches, and that the required linguistic knowledge is mainly that encoded in dependency relations and in argument structure.

In the recent past, statistical systems have gained considerable success in NLP. However, we can observe an increasing interest in *hybrid* approaches where statistical models are integrated with linguistic information. Applying such kind of knowledge can help to identify translational correspondences in a more efficient way.

Secondly, the meaningful improvement of performance in several NLP tasks determined by dependency-based formalisms applied in various treebanks motivates the investigation of the influence of knowledge encoded in dependency relations also with respect to alignment systems and MT. The choice of such a paradigm in this study is also dictated by the fact that dependencies can better represent linguistic phenomena typical of morphologically rich and free-word order languages; furthermore, the representation provided by dependencies shows, on the one hand, a higher degree of cohesion, compared to phrase-based representation, as demonstrated in Fox [14], and, on the other, it is closer to the semantic level, especially if enriched with argumental roles. In a cross-linguistic perspective, this aspect offers the possibility of having, in several cases, a more similar structural representation for a pair of sentences in different languages.

Neverthless, as for the alignment task in particular, dependencies may present a drawback consisting in the fact that for the appropriate identification and treatment of translational divergences in a tree pair, the only word level may not be satisfactory: a sub-sentential level is somewhat required. The hypothesis that will be explored in the study proposed herein is namely that such information could be covered by the knowledge on the predicative structure encoded in dependency substructures.

After an outline of recent projects on parallel treebanks (Sect. 2), this paper offers an overview of the treebank (Sect. 3), with a description of its content and the format used for the linguistic annotation; it also provides (Sect. 3) a detailed description and a tentative classification of the translational divergences encountered in the resource and the alignment approach we intend to pursue to properly handle such divergences.

2 Parallel Treebanks and Their Alignment

Over the recent years, several projects have been carried out on parallel treebanks, based both on constituency and dependency paradigms. Among them:

- Prague Czech-English Dependency Treebank (PCEDT)[1]: in its second release (2.0), it contains 2,312 documents from the English Penn Treebank-Wall Street Journal Section and its Czech translation, annotated in the Prague Dependency Treebank style. The corpus is 1:1 sentence-aligned and an additional automatic alignment on the node level—for each annotation layer—is also provided.
- Stockholm MULtilingual TReebank (SMULTRON)[2]: first developed by the Computational Linguistics Group at the Department of Linguistics of Stockholm University, and then maintained and enriched at the Institute of Computational Linguistics at Zurich University. The latest version (3.0) consists of around 2,500 sentences represented according to a constituency formalism and stored in TIGER-XML files. Parallel sentences are aligned on word and phrase level[3] with the Stockholm TreeAligner,[4] a graphical user interface which supports the manual alignment of parallel trees in the TIGER-XML format.
- Bulgarian–English Treebank [32]: a parallel treebank developed according to the principles of the Bulgarian HPSG resource grammar BURGER and the ERG resource grammar for the English counterpart. The treebank is automatically aligned on the word as well as the semantic level [31], using an approach that is inspired by the work on Minimal Recursion Semantics [7]. This work has been recently extended to include Portuguese [13].
- Copenhagen Dependency Treebanks (CDT)[5]: a collection of texts annotated on the basis of the dependency-based Discontinuous Grammar. Besides the monolingual Danish Dependency Treebank, the collection includes the Danish–English Dependency Treebank and other parallel annotated texts for German, Italian and Spanish.
- German–Georgian, German–Russian, German–Ukrainian (GRUG) parallel treebank[6]: a treebank containing 2,600 sentence pairs for each sub-corpus. This dataset is made of two types of resources: four monolingual treebanks (German, Georgian, Russian and Ukrainian), and four parallel treebanks (German–Georgian, German–Russian, German–Ukrainian, Georgian–Ukrainian). The parallel texts used comprise German sentences and their translations into Georgian and Russian languages. Similarly to SMULTRON, parallel trees are represented in the

[1] http://ufal.mff.cuni.cz/pcedt2.0/en/index.html.

[2] http://www.cl.uzh.ch/research/paralleltreebanks/smultron_en.html.

[3] Contrarily to work on statistical machine translation, phrase alignment in this work is intended as an alignment between linguistically motivated phrases.

[4] http://kitt.cl.uzh.ch/kitt/treealigner.

[5] http://code.google.com/p/copenhagen-dependency-treebank/.

[6] http://fedora.clarin-d.uni-saarland.de/grug/.

TIGER-XML format and manually aligned on the sentence, word and phrase level with the Stockholm TreeAligner.

This type of parallel resources is often based on the application to one or more languages of a format previously tested on a single language. The literature shows several examples of application to different languages of formats originally developed for a given language, by using the same features of the native format to address new linguistic phenomena encountered in the other languages. For instance, the format of the Prague Dependency Treebank (PDT), developed for Czech, has been afterwards applied to Arabic [15], while the Penn Treebank format has been applied e.g. to Chinese[7] and Arabic.[8]

Given the increasing importance of parallel aligned treebanks in many NLP tasks, several contributions were presented on the creation of automatic alignment systems, see e.g. [19, 34, 36]. They mainly deal with the alignment of parallel phrase structures and the process is often determined and regulated by the so-called *wellformedness* constraint, where (a) a node can only be linked once, and (b) an ancestor/descendant in the source tree should only be aligned to an ancestor/descendant of its counterpart in the target tree. Although the main goal of such works is to exploit alignment tools for MT, there were also few cases where the purpose was to make explicit the syntactic divergences between sentence pairs, as in Hearne et al. [16]. According to the authors, the major benefit from aligning syntactic structures, and phrase structures in particular, consists in the opportunity to infer translational correspondences between two substrings in the source and target side by allowing links higher up in the trees.

A relatively limited amount of contributions have described approaches based on some notion of dependency and argument structure. In Dyvik et al. [12], a general framework is proposed for the alignment of parallel trees represented according to the Lexical Functional Grammar principles. In this work, a different constraint is posed for the tree alignment: a pair of source and target words are considered as translational equivalents, and then aligned, (a) if the words are always (out of context) considered as translation of each other, and (b) if they occupy corresponding positions within the corresponding argument structure.

Works on alignment of deep syntactic structures in terms of dependency relations include, for example, that of Ding et al. [11], who developed an algorithm that uses parallel dependency structures to iteratively add constraints to possible alignments. In each iteration, the algorithm first trains the translation model to acquire the word-to-word translation probabilities, then choses the best alignment score for the remaining unaligned nodes based on a heuristic function, and re-estimates the translation model in the next iterations; an extension of such work is that of Ding and Palmer [10], who used a statistical approach to learn dependency structure mappings from parallel corpora, assuming at first a free word mapping, then gradually adding constraints to word level alignments by breaking down the parallel dependency structures into smaller pieces called *treelets*. Mareček et al. [22] proposed an

[7] http://www.cis.upenn.edu/~chinese/.

[8] http://www.ircs.upenn.edu/arabic/.

alignment system of the tectogrammatical layer of texts from the Prague Czech–English Dependency Treebank[9] with a greedy feature-based algorithm that exploits some measurable properties of Czech and English nodes in the corresponding tectogrammatical layers. Among these works, three in particular presented a common approach consisting in the creation of an initial set of word alignment which is then propagated to the other nodes in the source and target dependency trees using syntactic knowledge. Menezes and Richardson [23] presented a mapping algorithm of parallel trees in the so-called Logical Form, a graph which represents the relations among the most meaningful elements of a sentence (similarly to the tectogrammatical layer of the Prague Dependency Treebank); the algorithm uses a best-first search strategy starting from the nodes with the highest correspondences, and then moves outward from this initial set of aligned nodes using a set of alignment rules; similarly, in Ozdowska [29] a set of anchor points is created by means of co-occurrence counts, then the alignment is propagated with a set of heuristics based on syntactic dependencies, while in Ma et al. [21] a high-precision anchor set is obtained with the intersection of bidirectional word alignment through IBM models and the alignment propagation to other nodes is determined by a set of syntactic features.

Despite the specific differences which characterize the individual works, they all largely inspired our research, both in its theoretical foundations and in the system design.

3 ParTUT

ParTUT has been designed as a multilingual development of an Italian existing treebank, the Turin University Treebank[10] (henceforth TUT), i.e. the reference treebank for evaluation campaigns for Italian[11] (see also [1, 2]), on which the state of the art for parsing this language is currently defined.

The multilingual perspective was in the spirit of the TUT project from the beginning—as a small English sample corpus available in the TUT's web site shows. But the foundations of ParTUT have been laid only in 2009, when 200 sentences extracted from the JRC-Acquis multilingual parallel corpus (see below for references) were annotated in the TUT format for Italian and in the Easy format for French, within the context of a cooperation between the organizers of Evalita and those of the French parsing evaluation campaign Passage.[12] This small corpus, later annotated in the TUT format also for French and English, has been the core of ParTUT.

The strategy adopted in the development of ParTUT consists in focusing first on the annotation quality rather than on the treebank size. Thus we started from a limited amount of data to be annotated in a very detailed and checked way, and we

[9] http://ufal.mff.cuni.cz/pcedt2.0/.

[10] http://www.di.unito.it/~tutreeb.

[11] http://www.evalita.it/.

[12] http://atoll.inria.fr/passage/eval2.en.html.

progressively collected and integrated in the annotation scheme and in the alignment tools the hints coming from the experience gained to be exploited in the development of larger datasets. The same approach has been previously successfully applied within the Italian TUT project, where the parallel growth of the resource and of a rule-based parser allowed the development of a testbed for parsing that, though of limited size, led to performances that positively compare to those for English. On the one hand, this strategy based on the annotation of a small dataset exposes to the risk of getting random and skewed results; but, on the other hand, the fully automatic annotation of a larger resource may lead to a less tested and correct dataset, see e.g. the well-known criticisms about the Penn Treebank. The past experience in the development of TUT shows the suitability of the former strategy, and the usefulness in training and testing NLP tools also of small gold standard datasets. In the case of a parallel resource where we deal with the alignment issues from scratch, this approach is further motivated by the need of starting from a limited amount of very frequent syntactic structures and translation shifts to be considered as a model for dealing with other less frequent ones that will be found in larger datasets.

3.1 Data

The retrieval of appropriate texts for the development of ParTUT is an aspect, among the others that we mention below in the section, that has influenced the current composition of the treebank. Collecting parallel texts (i.e. that are in translational relation to each other) may not be so trivial; such texts are often protected by copyrights, or it could be necessary to create new translations from texts in a given source language (which is a time-consuming task in itself). Not surprisingly, among the most used resources in multilingual NLP, there are large collections of texts belonging to organizations such as the European Union, which makes available all its documents in all the official languages. In ParTUT as well we have taken advantage of this availability, also considering possible future enrichment of our collection with respect to languages not included in ParTUT for the time being.

ParTUT currently comprises around 89,000 tokens, with an average amount of 1,060 sentences per language, but a new release of the treebank will be available by the end of 2014. The texts of the collection currently available were gathered from:

- the *Creative Commons* open licence[13] (**CC**);
- the well-known and most commonly used Europarl parallel corpus[14] [18] (**Euro**);
- publicly available pages from Facebook website[15] (**FB**);
- the JRC-Acquis multilingual parallel corpus, i.e. the total body of the EU law[16] [33] (**JRC**);

[13] http://creativecommons.org/licenses/by-nc-sa/2.0.

[14] http://www.statmt.org/europarl/; the section used is `ep_00_01_17`.

[15] Namely the "Help" section, at https://www.facebook.com/help/345121355559712/.

[16] The section used is `jrc52006DC243`.

Table 1 Corpora and size of ParTUT

Language	Sentences	Tokens
English	1,068	27,632
French	1,049	30,971
Italian	1,077	30,585
Total	3,194	89,191

- the whole text of Universal Declaration of Human Rights[17] (**UDHR**);
- the Web Inventory of Translated Talks[18] [6] (**WIT3**).

The composition of the treebank is summarized in Tables 1 and 2.[19] Although to varying degrees, three of the sub-corpora belong to the legal domain (namely CC, JRC and UDHR). Choosing such texts, we benefitted from the expertise in the field of legal language processing acquired within the TUT project, where around 30 % of data are extracted from legal texts, i.e. the *Codice Civile* and the *Costituzione Italiana*. Being aware that analyses based on such kind of unbalanced material may affect the validity of the whole treebank, we extended the treebank (and we are currently working on its further extension) to other text genres, comprising debates of the European Parliament (EURO), instructions on how to create a Facebook account (FB) and multilingual transcriptions of talks from the TED Conferences (WIT3).[20]

As regards the development of the treebank, ParTUT is automatically annotated with the Turin University Linguistic Environment (TULE) [20], and then entirely manually corrected also exploiting hints from automatic check tools, extending the strategy applied in the case of Italian TUT also to the new resource. TULE implements a pipeline from tokenization to lemmatization and PoS tagging, to morphological analysis and dependency parsing (for a detailed description of the output format, see Sect. 3.2). Although in principle it supports linguistic analysis of several languages other than Italian (English in particular, but also French, Spanish, Catalan and Hindi), its output quality achieved satisfactory results mostly for Italian, as it has been extensively used in the development of TUT. For the development of ParTUT, TULE has been firstly tested on a small selection of English and French texts. This test phase entailed alternating steps of rule insertion and automatic analysis, until an output of acceptable quality was produced. Rule-insertion steps mainly included the enrichment of lexical knowledge, e.g. the insertion of new lexical entries (including proper Nouns, named entities, compounds and locutions), modifications in the suffix tables and new disambiguation rules for linguistic phenomena previously unseen in

[17] http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx.

[18] https://wit3.fbk.eu/; we retrieved the texts used for training of MT systems, downloaded from https://wit3.fbk.eu/mt.php?release=2012-02.

[19] As for the sentence count, we would like to clarify that some sub-corpora, especially the UDHR, are featured by the presence of short headings (e.g. 'Article 1') that we did not considered for calculating the average sentence length, even if they were treated as separate sentences according to the parser segmentation criteria.

[20] In general, considering the sources from which the texts of ParTUT have been retrieved, it can be assumed that they are not all original, but drafted in one or more languages and then translated into the others.

Table 2 Corpora and size of ParTUT

Corpus	Sentences	Tokens
CC_En	89	2,541
CC_Fr	102	3,208
CC_It	100	3,492
Euro_En	517	14,090
Euro_Fr	480	14,817
Euro_It	505	14,572
FB_En	114	1,736
FB_Fr	112	1,960
FB_It	115	2,000
JRC_En	180	5,611
JRC_Fr	179	6,902
JRC_It	181	6,753
UDHR_En	77	2,150
UDHR_Fr	77	2,401
UDHR_It	76	2,240
WIT3_En	91	1,504
WIT3_Fr	99	1,683
WIT3_It	97	1,528

Italian but occurring in English and French. The parser was then run for all the data of the parallel treebank.

The final step of the annotation consisted in the correction of the output of TULE for all the three languages. This task is usually performed by two independent annotators skilled in both TUT format and the annotated languages, and it is followed by the discussion of cases and phenomena not previously encountered by the annotators. While for Italian TULE achieves performances at the state of the art (i.e. around Labelled Attachment Score 90 %), it underperforms on English and French producing a larger amount of errors, despite the manual tuning of the system. Nevertheless, there are some main advantages in the application of TULE for all the three languages: the output is in the same and rich format typical of TUT, and a variety of tools are available for error detection, together with the guidelines collected during the development of TUT which proved useful in solving most of the disagreement cases. The guidelines have been then integrated with the new phenomena encountered.

3.2 Annotation Format

As mentioned in the previous section, ParTUT is a parallel dependency treebank annotated in compliance with the principles and using the same Part of Speech (PoS) tags and syntactic labels of the TUT format, applied to the Italian monolingual

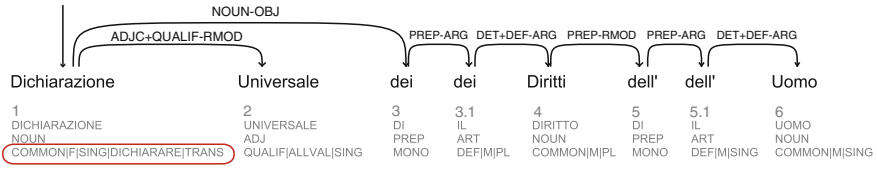

Fig. 1 Example of the Italian sentence "Dichiarazione Universale dei Diritti dell'Uomo" (*Universal Declaration of Human Rights*) annotated in the TUT format

treebank. This format represents surface-oriented projective dependency trees whose nodes are labeled with words, and whose arcs are labeled with the names of syntactic relations.

A typical sentence produced by TULE, and annotated according to TUT format specifications, is displayed as follows:

```
************** Frase HUMAN-RIGHTS-1 **************
1 Dichiarazione (DICHIARAZIONE NOUN COMMON F SING DICHIARARE TRANS) [0;TOP-NOUN]
2 Universale (UNIVERSALE ADJ QUALIF ALLVAL SING) [1;ADJC+QUALIF-RMOD]
3 dei (DI PREP MONO) [1;NOUN-OBJ]
3.1 dei (IL ART DEF M PL) [3;PREP-ARG]
4 Diritti (DIRITTO NOUN COMMON M PL) [3.1;DET+DEF-ARG]
5 dell' (DI PREP MONO) [4;PREP-RMOD]
5.1 dell' (IL ART DEF M SING) [5;PREP-ARG]
6 Uomo (UOMO NOUN COMMON M SING) [5.1;DET+DEF-ARG]
```

Observing the example[21] above, we can see that the native TUT format encodes, for each node (i.e. for each line in the file), the position of the token within the sentence, its word form, its lemma together with the PoS tag and morphological features (in round brackets), and the position of the governor with the relational label that links the node to its governor (in square brackets). Figure 1 shows the same sentence graphically drawn, and comprising all the linguistic information encoded in the annotation.

For what concerns, in particular, the dependency relations annotated in TUT, their labels may include two components[22] according to the following pattern:

$$morphoSyntactic-functionalSyntactic.$$

The main (and mandatory) component is the second one, specifying the syntactic function of the node in relation to its governor, i.e. whether the node is an argument (ARG), a modifier (MOD) or a more specialized kind of argument (e.g. OBJ) or modifier (e.g. RMOD for restrictive modifier). This component can be preceded by another one that specifies the morphological category (a) of the governing item, in

[21] In this paper, we report examples of sentences (or fragments of sentences) in all the languages involved. The glosses for non-English examples are then provided; they are intended as literal and do not necessarily correspond to the correct English expression.

[22] In the Italian TUT there is a third component (omitted here and in the current ParTUT annotation) concerning the semantic role of the dependent with respect to its governor.

case of arguments (e.g. PREP-ARG for the argument of a Preposition, like node 5.1 in Fig. 1), (b) of the dependent, in case of modifiers (e.g. PREP-RMOD for a prepositional restrictive modifier, like node 5 in Fig. 1). In some cases, the subcategory type of this additional component is also included (after the plus sign), as in DET+DEF-ARG, which should be read as argument of a definite Determiner, like node 6 in Fig. 1. Note that, in TUT, the root of a Noun group is the Determiner (if any), while the root of a prepositional group is the Preposition, as prescribed in the *Word Grammar* [17], which is the main reference theoretical framework for TUT.

Other characteristics of the TUT format are designed for maximizing the possibility of extraction of linguistic knowledge from the annotated material, and are motivated by the linguistic features of the languages on which it has been applied. For instance, compound Nouns and contracted forms are split into their components, with an associated node in the parse tree for each of them. This means, for example, that in the sentence in Fig. 1, the word "*dei*" (i.e. node 3 and 3.1), resulting from the contraction between the Preposition "*di*" (*of*) and the masculine plural Article "*i*" (*the*), is split in two distinct nodes for each of their components. The same happens for multi-word expressions, where each of their components is associated with a different node, although in this case they share the same lexical (i.e. lemma) and morpho-syntactic information.

Another relevant feature is that the format is oriented to an explicit representation of the predicate-argument structure, which is applied to Verbs, but also to Nouns and Adjectives; this means, for example, that in case of deverbal Nouns, their arguments are annotated as arguments of the corresponding verbs. Figure 1 also shows an example of nominalization of the transitive verb "*dichiarare*" ("*declare*") into the deverbal Noun "*dichiarazione*" ("*declaration*"); the realization of the derived direct object is thus marked by the Preposition "*del*" ("*of*") and annotated with the relational label NOUN-OBJ (while the corresponding verbal direct object would be marked as VERB-OBJ). A distinction is also drawn between modifiers and subcategorized arguments, and between surface and deep realization of any admitted argument (e.g. in case a verb undergoes a transformation from active to passive form).

Moreover, contrarily to most of dependency-based annotations, TUT format also exploits null elements, in order to deal with pro-drops, long-distance dependencies and elliptical structures, and to preserve the projectivity constraint. Null elements can be co-indexed with some word of the sentence (e.g. for gapping or long-distance dependencies), while non co-indexed null elements are mainly used for the representation of elliptical constructions, pro-drop subjects or other dropped arguments which play some role in the predicative structure of verbs.

The richness and flexibility of the TUT format have mainly driven its choice as the reference format for the new resource as well. But there is still one reason that made us lean on this option, that is the availability of conversion tools, created in parallel with the Italian treebank development, from TUT into other formats that are currently known as *de facto* standards. The first result of such conversion processes

was TUT-Penn,[23] i.e. the Penn Treebank format adapted to Italian characteristics. A conversion procedure has also been recently developed for the application of the Stanford Dependencies[24] [9] to two Italian monolingual treebanks, TUT and ISST–TANL, thus resulting in the creation of the Italian Stanford Dependency Treebank (ISDT) [3]. Moreover, in order to make TUT format adequate for the evaluation campaigns for parsing, as Evalita (cited above), the scripts needed for the generation of a CoNLL version of TUT have been developed, with a reduced set of relations and the typical organization in ten columns. According to the CoNLL requirements (see also [4, 26]), the TUT-like annotated sentence shown above displays as follows:

```
1   Dichiarazione   DICHIARAZIONE   NOUN    NOUN    COMMON|F|SING|DICHIARARE|TRANS  0   TOP _   _
2   Universale  UNIVERSALE  ADJ ADJ QUALIF|ALLVAL|SING  1   RMOD    _   _
3   dei DI  PREP    PREP    MONO    1   OBJ _   _
4   dei IL  ART ART DEF|M|PL    3   ARG _   _
5   Diritti DIRITTO NOUN    NOUN    COMMON|M|PL 4   ARG _   _
6   dell'   DI  PREP    PREP    MONO    5   RMOD    _   _
7   dell'   IL  ART ART DEF|M|SING  6   ARG _   _
8   Uomo    UOMO    NOUN    NOUN    COMMON|M|SING   7   ARG _   _
```

CoNLL is also the format used for our experiments on syntactic alignment (see next session).

4 Translation Shifts and Alignment Issues

As stated above, the usefulness of parallel corpora in translation studies and machine translation is strictly related to the availability of aligned data. The absence of a tool, among the existing ones, which was suitable for this purpose and that was compatible with the TUT format, led us to the development of a new system.

Since the system is still under development, it is not possible to provide full quantitative data on its performance. In this section, however, we intend to describe the theoretical basis on which we have relied for its design and the empirical evidence that supported the choice of our approach.

The primary aim of ParTUT project is to create a parallel resource where corresponding parts of a bitext are aligned by exploiting their structures. The focus of this section is thus on the alignment of data in the resource. We describe how data are aligned, starting from the sentence level, and the principles and criteria we adopted in detecting and representing the syntactic structures which convey the same meaning in different languages. A small final section is devoted to the description of the alignment tool developed in parallel with the resource and where the knowledge about the alignment is collected in rules.

---

[23] The TUTtoPenn converter can be downloaded at http://www.di.unito.it/~tutreeb/TUTtoPENNconverter/.

[24] http://nlp.stanford.edu/software/stanford-dependencies.shtml.

Table 3 Percentage of 1:1 correspondence at the sentence level, distributed per corpus and language pair

Corpus	EN-FR	FR-IT	IT-EN	Avg
CC	68.8	67.8	77.9	**71.5**
EURO	92.9	87.8	87.7	89.4
FB	96.5	93.8	**85.3**	91.8
JRC	96.4	92	98.8	**95.7**
UN	92.8	97.2	98.1	**96**
WIT3	70.5	88	**55.1**	71.2
Avg	86.3	87.7	83.8	85.9

4.1 The Sentence Level Alignment

The search for translational correspondences and divergences at a structural level always starts from a basic mapping at the sentence level. Sentences in ParTUT are automatically aligned with the Microsoft Bilingual Aligner [24].[25] Sentence alignment allows us to estimate the percentage of 1:1 correspondences in the treebank, as reported in Table 3. The overall percentage of 1:1 correspondence in ParTUT texts is 85.9 %. With respect to the language pairs, the highest correspondence is between Italian and French (87.7 %), while for the English–French pair figures reach approximately 86.3 %, and 83.8 % for the Italian–English pair. Nevertheless, it can also be observed that the percentage is strongly influenced by the nature of the data and not necessarily determined only by the languages themselves. As it can be seen in the table, while in most of the sub-corpora Italian and English report the higher percentage of 1:1 correspondence, this value is drastically lowered by the significant difference in the WIT3 and FB corpus. Because of their nature, JRC and UDHR have instead the highest percentage of matches on the sentence level, unlike, for example, the sentences from the CC corpus.

4.2 The Syntactic Alignment

As far as the alignment at the syntactic level is concerned, an explorative analysis of data confirmed us that the exact match between structures is relatively rare and correspondences of all nodes like those found e.g. in the sentence pair shown in Fig. 2 occur a very few times in our corpora: indeed, only 13 of such occurrences were found in the Italian–French pair, 17 in French and English and 37 in English and Italian. Nevertheless, the correspondence between full sentences or large part of them can be recovered as the result of the detection of systematic divergences in several cases. For this reason, we studied the various cases of divergences, which we

[25] A semi-automatic alignment has also been performed with LF Aligner (http://sourceforge.net/projects/aligner/).

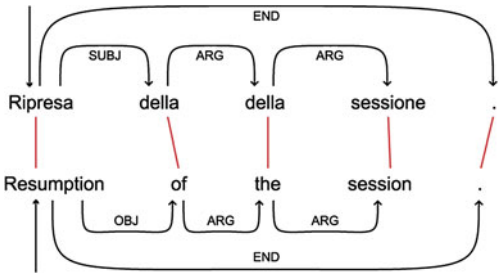

Fig. 2 Example of exact match and isomorphism between tree pairs

referred to as *shifts* [5, 8, 35], and we identified and classified those encountered in ParTUT in three main classes, each one involving morpho-syntactic, structural and semantic level respectively.

The first class is that of **Category shifts**, i.e. a divergence in the Part of Speech use between source and target text. It can be exemplified by the exploitation of a deverbal Noun rather than a Verb like in the following pair of fragments:

JRC_En#23[26]: ***Improving*** *the efficiency*[. . .]
JRC_Fr#24: ***L'amélioration de*** *l'efficacité* [. . .]
(*The improvement of the efficiency*)

The second class involves **Structural shifts**, and comprises all those cases where syntactic level is directly involved and affected from translator's choices or word order constraints. We then include the cases of:

- *discontinuous correspondences*

  UDHR_En#41: *Nor shall* ***a heavier penalty*** *be imposed* ***than*** *the one that was applicable* [. . .]
  UDHR_It#41: *Non potrà del pari essere inflitta* ***alcuna pena superiore a*** *quella applicabile* [. . .]
  (Cannot be likewise imposed any penalty heavier than the one applicable)

- *passivization/depassivization*[27]

  FB_En#9: *We don't* ***allow*** *accounts* [. . .]
  FB_It#11: *Gli account non* ***sono consentiti*** [. . .]
  (The accounts are allowed)

- *different syntactic realizations*, i.e. a wide sub-class that encompasses a variety of structural phenomena, e.g. light verb constructions and conflations of two items into a single one equivalent in meaning, paraphrases, locutions and idioms (such

[26] These labels are used to identify the treebank fragment we refer to in the examples: they indicate `section_language#sentencenumber`.

[27] Since in the ParTUT texts translation direction is unknown, we consider the two transformation strategies as counterparts one of each other and put them in the same subclass, while other works rather considered them as separate categories [8]. We applied the same principle even for the cases of addition/deletion, mentioned below.

as the one in the example below)

WIT_En#36: [. . .] *to bring that home* [. . .]
WIT_Fr#41: [. . .] *pour vous faire comprendre* [. . .]
(to make you understand)

- *function word introduction/elimination*, like in the case of nominal versus prepositional modification, such as the following:

JRC_En#55: [. . .] *environmental life cycle performance* [. . .]
JRC_It#55: [. . .] *prestazione ambientale del ciclo di vita* [. . .]

The third class, that of **Semantic Shifts**, includes cases where the level of meaning is somewhat affected, either by the addition or deletion of pieces of sentence, or by a translation sometimes too fuzzy compared to the source text. These cases are respectively referred to as:

- *addition/deletion*

UDHR_En#11: [. . .] *the respect for* ***and observance of*** *human rights and fundamental freedoms* [. . .]
UDHR_Fr#9: [. . .] *le respect* ***universel et effectif*** *des droits de l'homme et des libertées fondamentales* [. . .][28]
(the universal and effective respect of human rights and of fundamental freedoms)

- *mutation*

UDHR_En#28: *the right to* ***recognition as a person before the law***
UDHR_Fr#26: *le droit á la* ***reconnaissance de sa personalité juridique***
(*the right to the recognition of his legal personality*)

It should also be noted that establishing a clear-cut distinction for each kind of shift is a non-trivial issue, as multiple divergences often occur together.

Taking into account these shifts, and benefitting from the strengths of a dependency-based representation (e.g. with respect to word order issues and argument-structure orientation), we have designed a tool for their automatic alignment that explicitly resorts to such strengths in a number of ways.

4.3 ParTUTaligner: Algorithm and Results

The aligner is the place where the knowledge about the alignment is mainly formalized and stored. The approach we are currently applying is essentially rule-based (except for the first step), and this allows us to have more control over which information is actually relevant while handling translation shifts, and which is not.

Following the example of other similar approaches [21, 23, 29], we then start from a lexical mapping of the nodes in the tree pair, moving outwards to the unaligned

[28] In this example, in particular, we observe both additions and deletions while comparing the English sentence to the French version.

nodes thanks to the information available on syntactic structure. The algorithm, whose detailed description can be seen in Sanguinetti et al. [30], includes three main steps, that is one referring to the lexical level, one to syntactic dependencies, and one to deal with multiple alignment links.

The first step identifies lexical correspondences and stores them in lexical pairs; the mapping of source and target nodes is carried out using GIZA++ [27] in both translation directions. The two alignments (from source to target and from target to source) were finally symmetrized and only the word pairs in the intersection set were retained.

In the second step, starting from the lexical pairs obtained in the first one, correspondences between neighbouring nodes are detected and the respective relational structure is compared in parallel texts.

Finally, a third step has been recently introduced to find mappings between sets of nodes that are left unlinked in the previous steps. Such step underlies the notion of *catena* [28] (pl. *catenae*), which is a syntactic unit recently introduced in dependency framework in order to describe linguistic phenomena such as elliptical and discontinuous constructions, for which other syntactic notions could not be applied. A catena may involve any group of nodes of a dependency tree, provided that they are continuous with respect to dominance; we therefore decided to explore its use for alignment, especially when dealing, for example, with paraphrases, idioms or conflations, i.e. all those cases where multiple alignment links are needed in order to preserve the translation equivalence. We then attempted to extract the possible catenae from ParTUT and integrated their use in the system design in order to deal with multiple alignment links, both one-to-many or many-to-many, that may be required because of a shift of some kind.

For the aligner preliminary evaluation, we sampled 60 sentences from the different subcorpora of the Italian and English sections. The sample was manually aligned by two independent annotators and then considered as our reference corpus. We selected the sentences so as to include, for each one, cases of translation shifts that fall into at least one of the categories described above.

Both the alignment produced automatically and the one edited manually are represented in the NAACL format,[29] where each line is formed by four columns containing the information on:

```
sentence_no, src_node, trg_node, [S|P]
```

where `sentence_no` is the id of the source sentence, `src_node` and `trg_node` are the positions of the source and target node respectively, while the tag `[S|P]` identifies a Sure (`S`) or Possible (`P`) alignment. An example of manually aligned shift in a sentence pair, which also highlights the underlying syntactic structure, is shown in Fig. 3.

We compared the alignment output with the alignment in the reference sample and attempted to assess its intrinsic quality by using distinct Precision, Recall and

[29] http://www.cse.unt.edu/~rada/wpt; http://www.cse.unt.edu/~rada/wpt05.

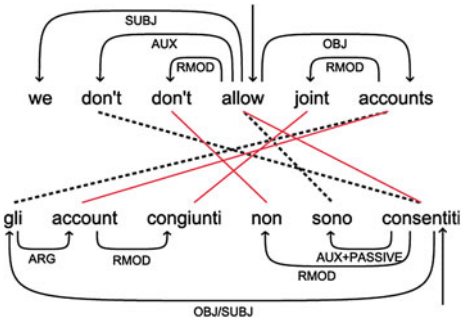

Fig. 3 Example of manual alignment of the sentences FB_En#9 and FB_It#11, also reported in Sect. 4.2. Sure links are drawn with *thick red lines* and Possible links with *grey dotted lines*. (Color figure online)

Table 4 Results of each alignment step: Precision, Recall and F measure for Sure (Ps, Rs, Fs) and Possible (Pp, Rp, Fp) alignments

Step	Ps	Rs	Fs	Pp	Rp	Fp
Intersection	57.3	69.3	62.4	76.3	39.1	51.2
With relations	58.2	76.3	62.2	71.1	43.8	53.5
With catenae	63.9	76.3	67.9	62.2	57	56.6

F-measure for Sure and Possible links, as reported in Table 4, and distinguishing the results obtained at each alignment step. The figures in the table show that major improvements are still needed for the system to be considered robust and efficient, especially if compared to past works that based the alignment process on some notion of dependency, such as [21], or [25].

As expected, the overall Precision score in Step 1 is higher than the scores obtained for the same measure in the next steps, though far higher for P links than for the S ones. Conversely, Recall score in Step 2 is higher for Sure links than for Possible ones.

Nonetheless, the interesting data observed is that the use of catenae contributes, although still not to a satisfying extent, to an improvement of the alignment system. This motivates further investigation of this notion and of its application in our research.

5 Conclusion and Future Work

In this paper, we presented preliminary results in the creation of ParTUT, a multilingual parallel treebank for Italian, English and French represented in the format of the Italian treebank TUT. TUT format proved to be a good candidate for the parallel annotation of three different languages, namely because of its dependency-based

representation and its focus on predicate-argument structure. These two factors in particular seemed crucial for the design of an alignment system that could properly put in correspondence parallel tree pairs, even when some translational divergence, or shift, occurs.

At the current stage, we oriented our efforts mainly to the development of a gold dataset featured by a high quality annotation, though of a limited size, following the same strategy successfully applied to the monolingual treebank TUT. According to this perspective, we designed the representation of a rich collection of cross-language shifts also formalized within a rule-based alignment system.

Several directions, however, are planned for this project to continue. First, a new extended release of the resource is expected by the end of the year, which includes new annotated texts. Second, though in parallel with the first point, we are working on the improvement of the alignment system, which by that time could then be tested on a larger dataset. Finally, following the tradition of TUT and previous experiments on ParTUT itself, we are working on the conversion into the Stanford typed dependencies, in order to improve the portability of the resource and its usability in a variety of NLP tasks.

References

1. Bosco C., Mazzei A.: The EVALITA dependency parsing task: from 2007 to 2011. In: Proceedings of Evalita 2011, Evaluation of Natural Language and Speech Tools for Italian. LNCS/LNAI, Springer (2012)
2. Bosco C., Mazzei A., Lavelli A.: Looking back to the EVALITA constituency parsing task: 2007–2011. In: Proceedings of Evalita 2011, Evaluation of Natural Language and Speech Tools for Italian. LNCS/LNAI, Springer (2012)
3. Bosco, C., Simi, M., Montemagni, S.: Converting Italian Treebanks: towards an Italian stanford dependency treebank. In: Proceedings of the ACL'13 7th Linguistic Annotation Workshop and Interoperability with Discourse (LAW&ID), Sofia, Bulgaria (2013)
4. Bucholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of CoNLL (2006)
5. Catford, J.C.: A Linguistic Theory of Translation: An Essay on Applied Linguistics. Oxford University Press, Oxford (1965)
6. Cettolo, M., Ghirardi, F., Federico M.: WIT3: a web inventory of transcribed talks. In: Proceedings of the 16th EAMT Conference, Trento, Italy (2012)
7. Copestake, A., Flickinger, D., Pollard, C., Sag, C.: Minimal recursion semantics: an introduction. Res. Lang. Comput. **3**(4), 281–332 (2005)
8. Cyrus, L.: Building a resource for studying translation shifts. In: Proceedings of Language Resources and Evaluation Conference (LREC'06), Genova, Italy (2006)
9. de Marneffe, M-C., Manning, C. D.: The stanford typed dependencies representation. In: Proceedings of the COLING'08 Workshop on Cross-Framework and Cross-Domain Parser Evaluation (CrossParser'08), Manchester, United Kingdom (2008)
10. Ding, Y., Palmer, M.: Automatic learning of parallel dependency treelet pairs. In: Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04) (2004)
11. Ding, Y., Gildea, D., Palmer, M.: An algorithm for word-level alignment of parallel dependency trees. In: The 9th Machine Translation Summit of the International Association for Machine Translation (2003)

12. Dyvik, H., Meurer, P., Rosén, V., De Smedt, K.: Linguistically motivated parallel parsebanks. In: Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8) (2009)
13. Flickinger, D., Kordoni, V., Zhang, Y., Branco, A., Simov, K., Osenova, P., Carvalheiro, C., Costa F., Castro, S.: ParDeepBank: multiple parallel deep treebanking. In: Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (2012)
14. Fox, H.J.: Phrasal cohesion and statistical machine translation. In: Proceedings of the ACL-02 conference on Empirical methods in Natural Language Processing (EMNL'02) (2002)
15. Hajič, J., Zemánek, P.: Prague Arabic dependency treebank: development in data and tools. In: Proceedings of NEMLAR the NEMLAR Conference on Arabic Language Resources and Tools (2003)
16. Hearne, M, Tinsley, J., Zhechev, V., Way, A.: Capturing translational divergences with a statistical tree-to-tree aligner. In: Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07) (2007)
17. Hudson, R.: Word Grammar. Blackwell, Oxford (1984)
18. Koehn P.: Europarl: A parallel corpus for statistical machine translation. In: Machine Translation Summit X, Phuket, Thailand (2005)
19. Lavie, A., Parlikar, A., Ambati, V.: Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In: Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation (SSST'08) (2008)
20. Lesmo, L.: The Turin University Parser at Evalita 2009. In: Proceedings of Evalita'09, Reggio Emilia, Italy (2009)
21. Ma, Y., Ozdowska, S., Sun, Y., Way, A.: Improving word alignment using syntactic dependencies. In: Proceeding of the Second ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2) (2008)
22. Mareček, D., Žabortský, Z., Novák, V.: Automatic alignment of Czech and English deep syntactic dependency tree. In: Proceedings of the 12th EAMT Conference (2008)
23. Menezes A., Richardson, S.D.: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: Proceedings of the Workshop on Data-driven Methods in Machine Translation at ACL-2001 (2001)
24. Moore, R.C.: Fast and accurate sentence alignment of bilingual corpora. In: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas: From Research to Real Users, Tiburon, California (2002)
25. Nakazawa, T., Kurohashi, S.: Bayesian subtree alignment model based on dependency trees. In: Proceedings of 5th Joint Conference on Natural Language Processing, Chiang Mai, Thailand (2011)
26. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007 (2007)
27. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. In: Computational Linguistics, vol .29(1). MIT Press, Cambridge (2003)
28. Osborne, T., Putnam, M., Gross, T.: Catenae: introducing a novel unit of syntactic analysis. In: Syntax, 15(4) (2012)
29. Ozdowska, S.: Using bilingual dependencies to align words in English/French parallel corpora. In: Proceedings of the ACL Student Research Workshop (2005)
30. Sanguinetti, M., Bosco, C., Cupi, L.: Exploiting catenae in a parallel treebank alignment. In: Proceedings of the 9th Language Resources and Evaluation Conference (LREC'14). Reykjavik, Iceland (2014)
31. Simov, K., Osenova, P., Laskova, L., Savkov, A., Kancheva, S.: Bulgarian-English parallel treebank: word and semantic level alignment. In: Proceedings of Recent Advances in Natural Language Processing (RANLP), Hissar, Bulgaria (2011)
32. Simov, K., Osenova, P.: Bulgarian-English treebank: desing and implementation. In: Linguist. Issues Lang. Technol. - LiLT **7**(14) (2012)

33. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga, D.: The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. In: Proceedings of Language Resources and Evaluation Conference (LREC'06), Genova (2006)
34. Tiedemann, J., Kotzé, G.: Building a large machine-aligned parallel treebank. In: Proceedings of the 8th International Workshop on Treebanks and Linguistic Theories (TLT'08) (2009)
35. Vinay, J.P., Darbelnet, J.: Comparative Stylistics of French and English. John Benjamins, Amsterdam and Philadelphia (1958)
36. Zhechev, V., Way, A.: Automatic generation of parallel treebanks. In: 22nd International Conference on Computational Linguistics (COLING 2008) (2008)

Comparing Named Entity Recognition on Transcriptions and Written Texts

Firoj Alam, Bernardo Magnini and Roberto Zanoli

Abstract The ability to recognize named entities (e.g., person, location and organization names) in texts has been proved as an important task for several natural language processing areas, including Information Retrieval and Information Extraction. However, despite the efforts and the achievements obtained in Named Entity Recognition from written texts, the problem of recognizing named entities from automatic transcriptions of spoken documents is still far from being solved. In fact, the output of Automatic Speech Recognition (ASR) often contains transcription errors; in addition, many named entities are out-of-vocabulary words, which makes them not available to the ASR. This paper presents a comparative analysis of extracting named entities both from written texts and from transcriptions. As for transcriptions, we have used spoken broadcast news, while for written texts we have used both newspapers of the same domain of the transcriptions and the manual transcriptions of the broadcast news. The comparison was carried on a number of experiments using the best Named Entity Recognition system presented at Evalita 2007.

Keywords Named entity recognition · Entity detection · Written texts · Automatic transcriptions

1 Introduction

The term *Named Entity* was first coined in the context of the Sixth Message Understanding Conference (MUC-6) [13], basically meaning anything that can be referred with a proper name. The MUC-6 Named Entity (NE) task addressed the automatic

B. Magnini · R. Zanoli
FBK-irst, via Sommarive 18, 38123 Povo (TN), Italy
e-mail: magnini@fbk.eu

R. Zanoli
e-mail: zanoli@fbk.eu

F. Alam (✉)
SIS Lab, Department of Information Engineering and Computer Science,
University of Trento, 38123 Povo (TN), Italy
e-mail: alam@disi.unitn.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_4

identification of names of people, organizations and geographic locations in a text. Since then *Named Entity Recognition* (NER) is a basic step in most Information Extraction tasks, aiming to detect and classify proper names that occur in texts. In addition to named entities, other kinds of entities have received attention in the research community, including temporal expressions (e.g., time, date), numeric expressions (e.g., money, percent) and bioinformatics (e.g., protein, DNA, RNA, genes).

During the last several years NER has became a relevant task in various application scenarios such as Information Retrieval, Question Answering, Summarization and Topic Detection. Moreover, NER systems are available in several languages and are also currently available with commercial applications. A typical NER system takes as input an unlabeled text, as for instance, the sentence "U.N. official Ekeus heads for Baghdad.", and produces as output the same text where all the occurrences of entities are annotated, as in "[ORG U.N.] official [PER Ekeus] heads for [LOC Baghdad].", where *Ekeus* is marked as a named entity of type PERSON and *Baghdad* has been recognized as a LOCATION. A relevant challenge of the NER task derives from the fact that the same named entity can refer to entities of different types. For example, *Washington* could be the name either of a PERSON or of a LOCATION.

Although research on NER is not new in Computational Linguistics, most of the attention has focused on English, and most of the works reported in the literature are based on the news domain (e.g., MUC-7, MET-1, MET-2, ACE, EVALITA-2007 and 2009). Research on NER on Italian started in 1998 [3], and it has then received great attention after the EVALITA[1] initiative, where activities have included the development of resources, guidelines, corpora and evaluation methodologies for NER. In the EVALITA context, NE recognition from written texts in Italian has achieved an F1 of 82.04 % (the EntityPro2007 system [29], developed by the HLT Unit, FBK), and the authors later designed another system in 2009 [30]. An interesting system presented in [18], which is based on structural re-ranking models and it often leads to improve the performance of NER.

In the last few years research interest has emerged for recognizing NE from transcriptions (speech transcribed using automatic speech recognition system). However, recent studies carried on in the evaluation campaigns (e.g., EVALITA 2011) [2] have shown that the recognition of NEs from transcriptions is still a challenging task. The reason is the current performance of Automatic Speech Recognition, and particularly, because of out-of-vocabulary words and the lack of NE markers (e.g., orthographic features) that are present on normal written texts.

In this paper, we propose a comparative analysis for the extraction of Named Entities, both from written texts and transcriptions. The main purposes is to provide empirical evidences about the factors that affect the gap of performance in the two situations, and to highlight which features are best suitable for reducing such a gap. To our knowledge, this is the first systematic investigation of *Named Entity Recognition* from written texts and transcriptions for Italian.

[1] http://www.evalita.it/.

The paper is structured as follows. In Sect. 2, we review the main literature on Named Entity Recognition. The main challenges about NER on transcriptions are presented in Sect. 3. Section 4 provides details about the EntityPro system, which we have used for our experiments. In Sect. 5, we report the data sets that we have used for this comparative study, which includes both written texts (I-CAB) and transcriptions (EVALITA-2011) in Italian. Section 6 reports the experiments on different data sets and configurations of EntityPro. Finally, Sects. 7 and 8, respectively, discuss the results of the experiments and conclude the paper.

2 Named Entities Recognition

Several approaches to NER have been investigated in the literature. While the early ones were mostly based on handcrafted rules, most recent approaches use some kind of supervised learning. In such approaches, typical learning algorithms include Support Vector Machines (SVMs), Hidden Markov Models (HMMs), Maximum Entropy and Conditional Random Fields (CRFs); classifiers are trained using an annotated corpus combined with different features, and the resulting classifier is then tested over a test data set.

A typical NER task requires that a mention of a named entity is both recognized and categorized according to a limited set of fixed categories. As an example, *Palo Alto* is an entity of type LOCATION, *Valentino Rossi* is a PERSON, and *Samsung* is an ORGANIZATION. The annotation style of the output of a NER system is typically based on the IOB2 format, where each token is classified according to a set of predefined categories, representing the fact that the token is either inside of an entity (notated with "I"), or it is at the beginning of an entity (notated with "B"), or it is outside of the entity (notated as "O"). The IOB2 annotation for the sentence "U.N. official Ekeus heads for Baghdad." would be [*B-ORG U.N.*] [*O official*] [*B-PER Ekeus*] [*O heads*] [*O for*] [*LOC Baghdad*].

2.1 Approaches

The identification and the extraction of relevant features is a crucial step of any NER system, which is usually designed based on statistics or machine learning techniques. In a typical NER system three different kinds of features [17] are employed: (i) *word level features* (e.g., token, case information); (ii) *list lookup features* (e.g., gazetteers); and (iii) *corpus or document level features* (e.g., document frequency). As for the use of such features, different NER evaluation campaigns [2, 25, 26], have basically given two options to participants: close-modality and open-modality. In close modality, no external resources are allowed for extracting features and for training the system. Such external resources typically include gazetteers, NE dictionaries, ontologies, external corpora, external classifier and even natural language

processing tools (e.g., TextPro[2] [20], GATE,[3] OpenNLP[4]) aiming to extract part-of-speeches, lemmas and chunks. On the other side, in the open-modality any type of external resources is allowed. While the exact definition of open and close-modality slightly varies depending on the specific evaluation campaign, the goal is to assess the impact of Word-level and corpus-level features particularly for resource scarce languages. These kinds of study also help to understand whether it would be worth to invest on developing external resources.

The *word-level features* include the words themselves (both unchanged and lower-cased), character n-grams with prefixes and suffixes, shapes (words capitalization patterns, hyphen, mixed case, ends with digits), punctuation and digit patterns. The digit patterns include cardinal, ordinals, dates, percentages, intervals, identifiers, roman-number and word with digits (e.g., 3M). Character n-grams provide some information about morphology, human profession that ends with "ist" (e.g., scientist, journalist), nationality and languages that ends with "ish" and "an" (e.g., Turkish, Spanish, Italian) and names (e.g., Italian first and last names end with 'O' and 'i') [4].

The *list lookup features* include dictionaries, gazetteers, list of peoples names, organization names, stop word list (usually implemented as BLACK-list that can never be an entity) and common abbreviations. A gazetteer is typically defined as a geographical dictionary that include list of place names along with geographical and geologic information. There are several approaches to match the candidate word with one of the existing list, including (i) Stripped match [5]; (ii) Fuzzy-Match with edit distance [6], and (iii) Soundex algorithm [22].

The *corpus*, or *document level features* include meta-information (e.g., news header, email header), word or phrase frequency and co-occurrence.

Overall, the most widely used features in current NER systems include tokens, parts-of-speech (POS), lemma, character prefixes-suffixes, syntactic chunk labels, gazetteers, list of names, bag-of-words and shape (case information e.g., upper case, lower case) features.

2.2 Evaluation Campaigns

Since the MUC-6 initiative on NER, there has been significant progress for named entity recognition on clean written texts (e.g., newspaper articles) across different languages. On the other side, named entity recognition on automatic transcriptions has started in 1998, sponsored by DARPA [21, 23] in the Hub-4 Broadcast News evaluation campaign, followed by several initiative and studies (see, among the others, [14, 19, 24, 28]).

[2] http://textpro.fbk.eu/.

[3] http://gate.ac.uk/.

[4] http://opennlp.apache.org/.

In the NER task at the DARPA Hub-4[5] information extraction evaluation campaign, Palmer [19] showed that they obtained an F-measure of 71–81 % on automatic transcriptions whereas 88 % on the reference transcriptions. The datasets for the Hub-4 IE evaluation was prepared by MITRE/SAIC and BBN, and consisted of one million words and 50 thousands Named Entities. The test set consisted of 32 thousands words and 1,800 Named Entities. Both datasets are a combination of American broadcast news (television and radio) from a range of dates between 1996 and 1998.

Recent progress on NER on transcriptions has been made in France, where a number of evaluation campaigns have been organized, including ESTER-2 [11], Quaero [10] and ETAPE-2011 [12]. The aim of ESTER-2 was to evaluate the segmentation, transcriptions and NE recognition from broadcast news and TV shows (e.g., entrainment, debates) comprising accented speech and non-news shows with spontaneous speech. The goal of the Quaero challenge was the extraction of a set of structured and extended Named Entities from automatic transcriptions of broadcast news, debates and talk shows. Following the series of ESTER, the goal of ETAPE-2011 was to include a wide variety of speech quality and more difficult challenges of spontaneous speech. The main focus of ETAPE-2011 was to foster a general-purpose transcription system for professional quality multimedia materials. Following the above mentioned experiences, a broadcast news NER task for Italian has been organized in the EVALITA-2011 context [2], following the NER series at EVALITA 2007 [25] and 2009 [26].

Based on such evaluation experiences, particularly MUC, CoNLL and ESTER, it is evident that the performance of a NER system is greatly affected by the Word Error Rate (WER) of the transcription. ASR systems make different kinds of errors, including word insertion, word deletion and word substitution. Several ideas have also been proposed to tackle these problems, such as explicitly modeling the ASR errors or using more hypothesis produced by the ASR system [9]. In addition, out-of-vocabulary words are relevant in NER, on top of the ASR errors. In order to reduce the WER in ASR, the typical approach is to built models using large corpora with the most frequent words. However, the transcriptions from this kind of ASR systems may not be helpful for the NER where unlikely events (e.g., proper names) are important. Two other issues have a great impact on NER performance: the insertion of erroneous proper names and spontaneous speech. In order to improve the quality of the NER, it has been proposed (see [9]) to remove speech disfluencies from the transcriptions. Moreover, to deal with these ASR errors in NER, possible approaches suggest to explicitly model ASR errors and to exploit a search space bigger than the 1-best hypothesis.

[5] http://www.itl.nist.gov/iad/mig/tests/bnr/1998/.

3 Extracting NEs from Written Texts and Transcriptions

Named Entity Recognition on written texts has made remarkable progress in recent years. However, recognizing NEs on automatic transcriptions is a much more challenging task, as transcriptions are degraded documents (i.e., they lack orthographic information). Transcriptions usually consist of either all uppercase or all lowercase texts, with no punctuation, which often produces a significant decrease in the tagger performance [1, 27]. Below we report an example of a news (i.e., *Bins on fire. Yet another case in Mercato in Naples. No to gag. Protest in the late afternoon in Naples against the law on phone tapping.*) with its automatic transcription.

- News. *Cassonetti in fiamme. Ennesimo caso a Mercato a Napoli. No al bavaglio. Protesta nel tardo pomeriggio a Napoli contro la legge sulle intercettazioni.*
- Transcription. *cassonetti in fiamme ennesimo caso a mercato a napoli no al bavaglio protesta nel tardo pomeriggio a napoli contro la legge sulle intercettazioni*

Mercato and *Napoli* in the news refer, respectively, to a neighborhood and a city and consequently they are NEs. The fact that their names start with an uppercase letter can be considered a good indicator that they are proper names and perhaps NEs, as they are. However, in the automatic transcriptions, this precious information is missing. As a consequence it is difficult for the NE tagger to decide whether or not *mercato* simply refers to a common name (e.g., the place where buyers and sellers meet for the sale of goods) or, instead, to a proper name (e.g., the neighborhood of the city). In addition, the token *no* in *no al bavaglio* in the transcription, could be interpreted by the NER as an abbreviation (e.g., NO, an Italian Province that is a NE), as it is close to the word *napoli* and there is no punctuation in between.

Other common issues with ASR transcriptions are due to out-of-vocabulary words (OOV) of the ASR systems. In fact, during the system development words that do not occur frequently (out-of-vocabulary) in the training corpora are typically removed from the ASR vocabulary to reduce the model size and complexity. These out-of-vocabulary words are often proper names, which can further confuse the NE tagger.

An example of automatic transcription is given in Fig. 1. The sentence (i.e., *Osvaldo Negra zoologist at the Tridentine Museum of Natural Sciences*), includes errors in terms of word recognition (e.g., *omologo* instead of *zoologo*; *diventino* instead of *Tridentino*), word segmentation (e.g., *nei gradi* instead of *Negra*), and in terms of word capitalization (e.g., initials for the words *Museo*, *Scienze* and *Naturali*). In particular, in the sentence of the example, these transcriptions errors directly involve the proper name of a person and of an organization.

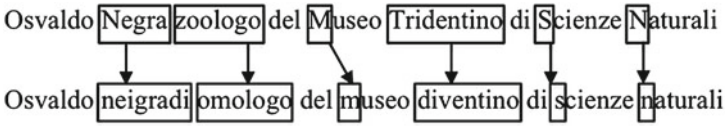

Fig. 1 An example of manual and automatic transcriptions, showing typical ASR errors on automatic transcription. The *upper row* is a manual transcription, whereas the *lower row* represents the automatic transcription

4 The EntityPro System

The experiments carried on for this paper are based on EntityPro [29], a NER system developed at FBK. EntityPro uses a rich set of linguistic features such as the Part of Speech (POS), and the occurrence of tokens in gazetteers to recognize NEs. The system architecture, as shown in Fig. 2, is based on YamCha.[6] It is a generic, customizable and open source text chunker implemented using SVM [7], which can be adapted to a number of NLP tasks. It allows to handle both static and dynamic features, and to define a number of parameters such as window-size, parsing-direction (forward/backward) and algorithms of multi-class classification problems (pair wise/one vs. rest). EntityPro has been trained to recognize four types of entities, namely: Geo-Political entity (GPE), Location (LOC), Organization (ORG) and Person (PER).

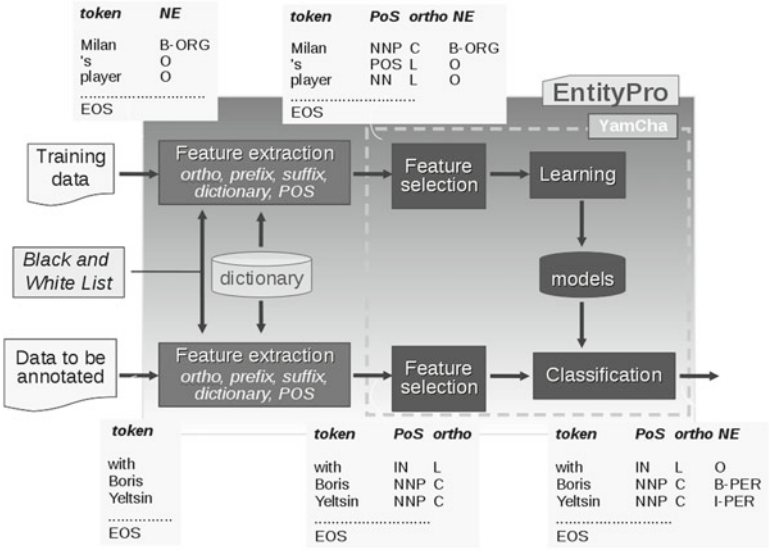

Fig. 2 EntityPro architecture

[6] http://chasen.org/~taku/software/yamcha/.

In addition, in order to allow easy optimization, EntityPro is provided with white and black lists of entities.

4.1 System Architecture

EntityPro is distributed with the following default configuration of features: the word itself, both unchanged and lowercased, its Part of Speech, prefixes and suffixes (1, 2, 3, or 4 characters at the start/end of the word); orthographic information (e.g., capitalization and hyphenation), collocation of bigrams (36,000 bigrams from Italian newspapers ranked by Mutual Information value); gazetteers of generic proper nouns extracted from the Italian phone-book and from Wikipedia (154,000 proper names), from various sites about Italian cities, (12,000), Italian and American stock market (5,000 organizations) and Wikipedia geographical locations (3,200); moreover a list of 4,000 proper nouns extracted from a sport newspaper (Gazzetta dello Sport, year 2004). Each of these features is extracted for the current, previous and following words. All the above features are *static* features, as opposed to *dynamic* features, which are decided dynamically during the tagging process. As for dynamic features the tagger uses the tag of the three tokens preceding the current token.

The tagger allows for two main operations, i.e., training the system and annotating a data set. Both operations are highly configurable as it is possible to: (i) select a certain feature set; (ii) choose different parameter values of the learning algorithm; (iii) provide the tool with black lists (i.e., words that the tagger must not recognize as NEs) and white lists of entities (words that the tagger has to recognize as NEs).

4.2 System Performance

The system was evaluated at Evalita 2007, where both the development and test data were part of the Named Entity task and external resources were allowed. EntityPro was configured by splitting the development set randomly into two parts: a data set for training (92,241 tokens) and a data set for tuning the system (40,348 tokens). The resulting best configuration was tested on the test set. EntityPro scored as the best system (evaluation based on exact match), reporting an accuracy of 82.14 (in terms of F1 measure) and of 74.07 without external resources (e.g., gazetteers).

5 Experimental Data Sets

To compare the NER performance on both written texts, manually and automatically transcribed broadcast news two different data sets were used, reported in the following sections.

5.1 Transcription Data Sets

This is the EVALITA-2011 NER dataset [2, 15], consisting of 20 broadcast news with a total of ten hours of transmission. Five hours were used for training and the other five hours for evaluation. The corpus was first manually transcribed and then manually annotated with NEs by three expert annotators. In addition, the same broadcast news were transcribed automatically by a state-of-the-art ASR system [8], with case restoration (for example *new york is the most populous city in the united states* becomes *New York is the most populous city in the United States* after case restoration).

As for the performance of the ASR system, the Word Error Rate (WER) on the training set is 16.39 and 17.91 % on the test set. The WER on named entities only is about 18.31 %, computed by first aligning both manual and automatic transcriptions and then comparing the tokens of each transcription containing NEs.

In the rest of the paper, we will refer to the transcribed data sets as *Evalita-train-manual*, i.e., training set of manually transcribed broadcast news, *Evalita-train-asr*, i.e., training set of automatically transcribed broadcast news, *Evalita-test-manual*, i.e., test set of manually transcribed broadcast news, and *Evalita-test-asr*, i.e., test set of automatically transcribed broadcast news. The distribution and the quantitative statistics of Named Entities over such datasets are given in Tables 1 and 2.

5.2 Written Text Data Sets

As for written texts, we used I-CAB (the Italian Content Annotation Bank) [16], which contains written news stories taken from different sections (e.g., News, Economic, Cultural, Local and Sports) of the local Italian newspaper L'Adige. I-CAB training set (I-CAB-train) consists of 525 news stories and the number of tokens is

Table 1 Statistics of transcriptions

	Train set	Test set
Broadcast news	10	10
Hours of transmission	5	5
Tokens	42,595	36,643

Table 2 Annotation statistics of the transcriptions

	Train set	(%)	Test set	(%)
GPE	747	38.82	572	39.46
LOC	105	5.46	88	5.17
ORG	618	32.12	527	30.94
PER	454	23.60	416	24.43
Total	1,924		1,703	

Table 3 Statistics of the written texts (I-CAB)

	Train set	Test set
News stories	525	180
Sentences	11,227	4,136
Tokens	212,478	86,419
Average tokens per news story	404.72	480.10

Table 4 Annotation statistics of the written texts (I-CAB)

	Train set	(%)	Test set	(%)
GPE	2,813	24.66	1,143	23.02
LOC	362	3.17	156	3.14
ORG	3,658	32.06	1,289	25.96
PER	4,577	40.11	2,378	47.88
Total	11,410		4,966	

212,478, while the test set (I-CAB-test) consists of 180 news stories with a total of 86,419 tokens. Tables 3 and 4 show the statistics of the corpus.

6 Experiments

We have carried out three experiments, on the three datasets reported in Sect. 5. Specifically, one experiment on the automatic transcription dataset (Sect. 6.2), one on the manually transcribed dataset (Sect. 6.3), in order to estimate the impact of the Word Error Rate of the ASR system, and finally on the ICAB written dataset (Sect. 6.4), in order to compare the performance of the NER on written texts and transcriptions.

For each experiment we have also investigated the importance of using external resources (e.g., resources like list of proper names and tools like a POS tagger). For this purpose we used EntityPro (see Sect. 4) with three different configurations: (i) with *word-level features* only, i.e., tokens with their prefixes and suffixes (i.e., Three characters at the start/end of the token); (ii) with *additional features*, i.e., POS of the tokens, lemma and chunk, and those features obtained by exploiting the list of proper names available with EntityPro; (iii) with *additional data set*, i.e., the I-CAB-train data set that will be used to enlarge the data set of the transcriptions. Figure 3 reports an example of the EntityPro configuration.

Pos	Token	POS	Chunk	Orth	PER	ORG	LOC	GPE	ETY	BLACK	Tag
-2	Cambia	VI	B-VX	C	O	O	O	O	O	O	O
-1	il	RS	B-NP	L	O	O	O	O	O	O	O
0	consiglio	SS	I-NP	L	O	B-ORG	O	O	B-ETY	O	B-ORG
1	di	E	O	L	O	I-ORG	O	O	I-ETY	O	I-ORG
2	amministrazione	SS	B-NP	L	O	I-ORG	O	O	I-ETY	O	I-ORG

Fig. 3 The default configuration of EntityPro. In the figure, static features are presented with the *dashed lined box* and the dynamic features, which are decided dynamically during the tagging are represented with the *solid lined box*

6.1 Evaluation Procedures

The performance of the system has been measured in terms of Precision, Recall and F1 measure, as reported by the CoNLL scorer.[7] *Precision* is the proportion of the correct positive predictions, which is computed as the ratio between the number of NEs correctly identified and the total number of NEs identified by the system as shown in Eq. 1. *Recall* is the proportion of positive cases recognized by the system and is computed as the ratio between the number of NEs correctly predicted and the total number of NEs that the system was expected to recognize as shown in Eq. 2. *F-measure* is the weighted harmonic mean of *Precision* and *Recall* computed using Eq. 3.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \tag{3}$$

where, TP, TN, FP and FN are true positive, true negative, false positive and false negative.

Baseline Results: As in the CoNLL-2002 Shared Task a baseline was produced by a system that only identifies entities in the test set, which have a unique class in the training data. In Table 5 we report the baseline results for the manual transcriptions and written texts data sets. Specifically, with Evalita-train-manual and Evalita-test-asr, respectively, as training and test sets, the baseline F1 is 49.98. While with Evalita-train-manual and I-CAB-train as training set and Evalita-test-asr as test set the F1 is 31.11. On the other hand, the baseline F1 calculated on the I-CAB data set is 36.85.

Another crucial aspect to be taken into account for our comparative evaluation is the results obtained by the best system at Evalita-2011. In that task, the 'closed' modality required to use the data distributed by the organizers and no additional resources (i.e., NE dictionaries, Wikipedia and complex NLP toolkits such as TextPro or OpenNLP) were allowed for training and tuning the system. However, the use of

Table 5 Baseline results for the manual transcriptions and written texts

Training set	Test set	Pr	Re	F1
Evalita-train-manual	Evalita-test-asr	73.76	37.80	49.98
Evalita-train-manual + I-CAB-train	Evalita-test-asr	28.80	32.54	31.11
I-CAB-train	I-CAB-test	40.29	33.95	36.85

The results are reported in terms of Precision (Pr), Recall (Re) and F1 measure

[7] http://www.clips.ua.ac.be/conll2002/ner/bin/conlleval.txt.

basic tools like a POS tagger was possible. In contrast to that modality, in the 'open' modality the use of any type of supplementary data was allowed. As regards the 'closed' modality the reported accuracy of the best system is of 60.98 in terms of F1, whereas 63.56 is the F1 obtained by the same system in the 'open' modality.

6.2 Experiments on Automatic Transcriptions Data Set

As already mentioned, orthographic information is one of the most important source of information for NER, and when not available a severe performance degradation might occur. To measure the impact of orthographic information, two different groups of experiments were conducted on the automatic transcriptions. One experiment was done on the transcriptions as produced by the ASR system using the Evalita data. Then, as the ASR performs case restoration, in the second experiment we removed the case of the tokens.

Automatic Transcriptions Without Case Information: In this section, we report the results on the experiments made by considering the lowercased transcriptions produced automatically by the ASR system. We used the Evalita-train-asr and the Evalita-test-asr, respectively, as training and test sets. Both the data sets were made lowercase and no punctuations were present. We conducted two different experiments, with and without the use of external resources.

Without External Resources: EntityPro was configured to use the *word-level features* only: they include the token itself and its prefixes and suffixes and no external resources were exploited. Table 6 shows the experimental results.

With External Resources: In these experiments, in addition to *word-level features*, *additional features* and *additional data set* were used to train EntityPro. Table 7 shows the results of these experiments.

Automatic Transcriptions with Case Information: This section focuses on the experiments using the transcriptions where the case of the tokens has been restored by the ASR system. The Evalita-train-asr and the Evalita-test-asr data sets were used

Table 6 Results on Evalita-test-asr lowercased, without using external resources

Category	Pr	Re	F1
Overall	73.60	43.56	**54.73**
GPE	82.20	64.97	72.58
LOC	59.26	34.41	43.54
ORG	58.28	32.65	41.85
PER	79.22	26.87	40.13

Table 7 Results on Evalita-test-asr, lowercased, using *additional features* (Feature), *additional data set* (Data Set) and using *additional features* in conjunction with *additional data set* (Feature + Data Set)

Category	Feature			Data set			Feature+Data set		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Overall	70.56	57.92	63.62	71.46	55.25	62.32	67.93	61.12	64.35
GPE	79.12	77.69	78.40	86.14	73.50	79.32	81.92	77.99	79.91
LOC	70.69	44.09	54.30	58.82	43.01	49.69	62.12	44.09	51.57
ORG	59.22	42.30	49.35	55.28	40.82	46.96	56.74	48.42	52.25
PER	66.86	50.22	57.36	68.12	48.02	56.33	59.86	54.85	57.24

Table 8 Results on Evalita-test-asr without using external resources

Category	Pr	Re	F1
Overall	71.61	49.03	**58.21**
GPE	79.97	68.71	73.91
LOC	65.52	40.86	50.33
ORG	59.94	36.92	45.69
PER	69.20	36.12	47.47

without any preprocessing. Similar to the previous case, these experiments were also done with and without considering the external resources.

Without External Resources: EntityPro was configured to use only *word-level features*. Table 8 shows the results.
With External Resources: Table 9 reports the results when *additional resources* were added to the basic *word-level features* and when *additional data set* were used in conjunction with Evalita-train-asr to train EntityPro.

Table 9 Results on Evalita-test-asr using *additional features* (Feature), *additional data set* (Data Set) and using *additional features* in conjunction with *additional data set* (Feature + Data Set)

Category	Feature			Data set			Feature+Data set		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Overall	71.44	59.18	**64.73**	64.58	56.56	60.30	67.96	61.69	64.67
GPE	79.33	78.14	78.73	82.91	73.35	77.84	83.25	78.89	81.01
LOC	72.73	43.01	54.05	59.38	40.86	48.41	70.97	47.31	56.77
ORG	61.90	45.83	52.67	54.89	45.83	49.95	58.41	48.98	53.28
PER	67.16	50.44	57.61	50.35	47.80	49.04	55.51	54.41	54.95

6.3 Experiments on Manual Transcriptions Data Set

The motivation of this experiment was to understand the impact of the Word Error Rate in ASR transcriptions to recognize Named Entities. In fact, manual transcriptions both contain the case information and do not contain errors, given that they were produced manually. In this scenerio, the case of the tokens has been restored by the human transcribers. For the experiments Evalita-train-manual was used as a training set, whereas Evalita-test-manual and Evalita-test-asr were used as the test sets.

Without External Resources: For these experiments we used *word-level features* only. Table 10 shows the results, with both manual and automatic transcriptions.
With External Resources: In this case, we used *additional features* to enlarge the features set. The results are given in Table 11.

Table 12 shows the results obtained by combining *additional data set* with Evalita-train-manual and then evaluated the system on the Evalita-test-manual and Evalita-test-asr data sets.

Table 13 reports the results where both *additional features* and *additional data set* were used.

Table 10 Results on Evalita-test-manual and Evalita-test-asr without using external resources

Category	Manual transcription			Automatic transcription		
	Pr	Re	F1	Pr	Re	F1
Overall	76.34	73.56	74.92	58.86	50.00	54.07
GPE	81.59	82.83	82.21	72.14	73.65	72.89
LOC	78.95	61.64	69.23	60.00	35.48	44.59
ORG	64.58	59.16	61.75	44.69	26.53	33.29
PER	80.47	78.41	79.43	48.27	46.04	47.13

Table 11 Results on Evalita-train-asr and Evalita-test-asr using the additional features

Category	Manual transcription			Automatic transcription		
	Pr	Re	F1	Pr	Re	F1
Overall	80.43	79.52	**79.97**	63.99	59.58	61.71
GPE	82.47	88.72	85.48	75.41	81.74	78.45
LOC	77.78	76.71	77.24	65.62	45.16	53.50
ORG	70.81	61.59	65.88	57.18	39.15	46.48
PER	87.34	86.89	87.11	51.68	54.19	52.90

Table 12 Results on Evalita-test-manual and Evalita-test-asr using *additional data set*

Category	Manual transcription			Automatic transcription		
	Pr	Re	F1	Pr	Re	F1
Overall	77.79	75.22	76.48	61.37	55.87	58.49
GPE	85.66	83.50	84.57	80.81	74.40	77.47
LOC	78.57	60.27	68.22	61.02	38.71	47.37
ORG	63.91	61.37	62.61	51.11	42.86	46.62
PER	81.49	81.49	81.49	45.86	47.58	46.70

Table 13 Results on Evalita-test-manual and Evalita-test-asr with *additional features* and *additional data set*

Category	Manual transcription			Automatic transcription		
	Pr	Re	F1	Pr	Re	F1
Overall	81.40	80.91	**81.16**	63.58	60.72	62.12
GPE	85.85	89.90	87.83	81.07	79.49	80.27
LOC	71.83	69.86	70.83	67.80	43.01	52.63
ORG	68.07	64.46	66.21	53.72	46.94	50.10
PER	91.01	88.43	89.70	49.18	53.08	51.06

6.4 Experiments on Written Text Data Set

To compare the results on the written texts and transcriptions, and considering the fact that I-CAB-train is much larger than Evalita-training set, we used only a small portion of the original I-CAB-train (about 41 K tokens) to train EntityPro. I-CAB-test was instead used as the test set.

Without External Resources: For these experiments we only used *word-level features*. Table 14 shows the experimental results.

With External Resources: We used both *additional features* and *additional data set* to train the EntityPro. The experimental results are reported in Table 15. Differently to the other experiments on transcriptions, where *additional data set* (i.e., I-CAB-train)

Table 14 Results on written texts (I-CAB-test) without using external resources

Category	Pr	Re	F1
Overall	67.42	56.09	61.24
GPE	69.75	65.41	67.51
LOC	44.19	12.18	19.10
ORG	49.55	42.98	46.03
PER	77.15	61.61	68.51

I-CAB-train small set (41K tokens) was used for training

Table 15 Results on written texts (i.e., I-CAB-test) using *additional features* (Feature), *additional data set* (Data Set) and using *additional features* in conjunction with *additional data set* (Feature+Data Set)

Category	Feature			Data set			Feature+Data set		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Overall	80.79	73.52	76.98	76.11	71.73	**73.85**	82.78	79.02	80.86
GPE	81.56	82.41	81.98	75.09	75.68	75.38	84.65	83.46	84.05
LOC	51.47	22.44	31.25	65.93	38.46	48.58	66.67	50.00	57.14
ORG	67.69	58.18	62.58	65.07	61.29	63.12	70.52	68.11	69.30
PER	87.93	80.91	84.28	83.09	77.67	80.29	89.47	84.69	87.02

was used in conjunction to the transcriptions training set, in this case only *additional data set* was used.

7 Discussion

In this section, we discuss the results of the experiments presented in Sect. 6. As a first aspect of our comparative study, we investigated the relevance of orthographic information through a number of experiments on automatic transcriptions, with and without case information. A quick look at the results seems to confirm that orthographic information is relevant: Table 6 reports 54.73 of F1 measure for the experiment that does not use the orthographic information and 58.21 (see Table 8) when this information is used. However, when we provide the NE tagger with external resources, orthographic information loses part of its relevance (i.e., +0.38 with the orthographic information).

The evaluation of the impact of the ASR errors was conducted by additional experiments on both automatic and manual transcriptions. The best F1 on automatic transcriptions is 64.73 (see Table 9), whereas the F1 for manual transcriptions is 81.16 (see Table 13). The difference in the accuracy is in line with the Word Error Rate of the ASR component on the NEs (i.e., 18.31 %), which can be considered as an upper bound, as the NEs that are not transcribed correctly by the ASR might not be recognized by the NE tagger.

Concerning the lack of punctuation in the transcriptions, from our experiments this phenomenon seems not to be so relevant. Specifically, we compared the NE tagger on the manual transcriptions containing the orthographic information, but not the punctuation, and with written texts containing both orthographic information and punctuation. Using a training data set of the same dimension for both the transcriptions and written texts, we obtained F1 79.97 for the manual transcriptions (see Table 11) and 73.85 for written texts (see Table 15). However, this observation is based on the experiments done considering two different data sets whereas a more precise comparison would require to use the same data set to exclude that the results

can be affected by the characteristics of the data sets rather than the punctuation information. In the future, we will provide more details about the experiments that we are thinking of doing to evaluate the impact of the punctuation to recognize the NEs.

The experiments with additional features (e.g., those generated by a POS tagger and by list of proper names) show a significant increment of the accuracy on all the data sets. As for the automatic transcriptions without orthographic information, we think that the main contribution is due to the list of proper names rather than the POS tagger or the lemmatizer. In fact, orthographic information is much more relevant for a POS tagger, whereas the accuracy of the other tools (i.e., the lemmatizer and chunker) depends on the accuracy of the POS tagger.

Finally, an interesting fact is that when we consider that at EVALITA-2011 participants have used manual transcriptions for developing systems that have then been evaluated on automatic transcriptions. According to our experiments, a better solution may consist in using the automatic transcriptions (i.e., the transcription as a training data set) to train the NE taggers too. The hypothesis is that, in order to optimize the performance of the taggers, the training data should be as much close to the evaluation set as possible. This is also confirmed by the experiments we carried on external data sets. In fact, when we tried to add written texts to the transcriptions data as training, in most of the cases we only had a small increment of the performance, and in all the other cases a decrease was noticed. The same result has been obtained by different baseline results on the same data sets. In contrast, when the written texts were enriched with data of the same type, the increment in accuracy was more consistent (i.e., +12.61).

8 Conclusion and Future Work

We have presented a study aiming at comparing Named Entities extraction on written text and on transcription data sets. To the best of our knowledge this is the first comparative study on the Italian language. To make this comparison more meaningful, we have used both comparable data sets (news of the same domain) and we run the same state-of-the-art entity tagger, EntityPro, on both of the data sets. As a first result, we have collected empirical evidences that the output of the ASR system contains recognition errors and presents missing information, such as orthographic information and punctuation, may reduce the performance of NER.

According to the initial intuition, the experiments show that performance on written texts is higher than automatic transcriptions mainly due to the ASR transcription errors. A less expected result is observed on capitalization information. Although it is an important source of information in the “no external resource” configuration, however, it is much less relevant as a discriminative feature when external resources are employed. Additionally, missing punctuation seems not to impact at all on the performance, which we observe by comparing the results on the transcriptions that do not contain the punctuation and written texts containing it. As a future work, to

exclude that this difference might be due to the data sets rather than punctuation, we are going to do some experiments using a single data set and removing and not removing the punctuation from it. Given that the transcriptions do not contain punctuation we are planning to use the written texts as a data set for this kind of experiments.

As for other future work, we intend to further investigate the impact of specific linguistic annotations (e.g., POS tags), as well as of specific feature combinations. A second aspect which would deserve future work is a better understanding of the relations between the errors of the ASR and the accuracy of the NER. Particularly, we would like to investigate the situation occurring when words that are not NEs in the broadcast news, are changed and transcribed as they were NEs, as when *il contropiede* is erroneously transcribed as *Bill Condon*, the famous American screenwriter.

References

1. Alam, F.: Named entity recognition on transcription using cascaded classifiers. In: Working Notes of EVALITA 2011 (2012)
2. Bartalesi Lenzi, V., Speranza, M., Sprugnoli, R.: Named entity recognition on transcribed broadcast news-guidelines for participants (2011)
3. Black, W.J., Rinaldi, F., Mowatt, D.: Facile: description of the NE system used for MUC-7. In: Proceedings of the 7th Message Understanding Conference (1998)
4. Chowdhury, M.F.M.: A simple yet effective approach for named entity recognition from transcribed broadcast news. In: Evaluation of Natural Language and Speech Tools for Italian, pp. 98–106. Springer, Berlin (2013)
5. Coates-Stephens, S.: The analysis and acquisition of proper names for the understanding of free text. Comput. Humanit. **26**(5–6), 441–456 (1992)
6. Cohen, W.W., Sarawagi, S.: Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods. In: Proceedings of the 10th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining, pp. 89–98. ACM (2004)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
8. Falavigna, D., Giuliani, D., Gretter, R., Lööf, J., Gollan, C., Schlüter, R., Ney, H.: Automatic transcription of courtroom recordings in the JUMAS project. In: 2nd International Conference on ICT Solutions for Justice, pp. 65–72. Skopje, Macedonia (2009)
9. Favre, B., Béchet, F., Nocéra, P.: Robust named entity extraction from large spoken archives. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 491–498 (2005)
10. Galibert, O., Rosset, S., Grouin, C., Zweigenbaum, P., Quintard, L.: Structured and extended named entity evaluation in automatic speech transcriptions. In: IJCNLP, pp. 518–526 (2011)
11. Galliano, S., Gravier, G., Chaubard, L.: The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In: Interspeech, vol. **9**. pp. 2583–2586 (2009)
12. Gravier, G., Adda, G.: Evaluation plan ETAPE 2011. http://www.afcp-parole.org/etape-en.html (2011)
13. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: COLING, vol. **96**, pp. 466–471 (1996)
14. Kubala, F., Schwartz, R., Stone, R., Weischedel, R.: Named entity extraction from speech. In: Proceedings of DARPA Broadcast News Transcription and Understanding Workshop, pp. 287–292. Citeseer (1998)

15. Lenzi, V.B., Speranza, M., Sprugnoli, R.: Named entity recognition on transcribed broadcast news at Evalita 2011. In: Evaluation of Natural Language and Speech Tools for Italian, pp. 86–97. Springer (2013)
16. Magnini, B., Pianta, E., Girardi, C., Negri, M., Romano, L., Speranza, M., Bartalesi Lenzi, V., Sprugnoli, R.: I-CAB: the Italian content annotation bank. In: Proceedings of LREC, pp. 963–968 (2006)
17. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvist. Investig. **30**(1), 3–26 (2007)
18. Nguyen, T.V.T., Moschitti, A.: Structural reranking models for named entity recognition. Intell. Artif. **6**(2), 177–190 (2012)
19. Palmer, D.D., Burger, J.D., Ostendorf, M.: Information extraction from broadcast news speech data. In: Proceedings of the DARPA Broadcast News Workshop, pp. 41–46. Citeseer (1999)
20. Pianta, E., Girardi, C., Zanoli, R.: The textpro tool suite. In: LREC (2008)
21. Przybocki, M.A., Fiscus, J.G., Garofolo, J.S., Pallett, D.S.: 1998 hub-4 information extraction evaluation. In: Proceedings of DARPA Broadcast News Workshop, (Herndon, Va, USA), pp. 13–18 (1999)
22. Raghavan, H., Allan, J.: Using soundex codes for indexing names in ASR documents. In: Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004, Association for Computational Linguistics, pp. 22–27 (2004)
23. Robinson, P., Brown, E., Burger, J., Chinchor, N., Douthat, A., Ferro, L., Hirschman, L.: Overview: information extraction from broadcast news. In: Proceedings of DARPA Broadcast News Workshop, pp. 27–30 (1999)
24. Sandrini, V., Federico, M.: Spoken information extraction from Italian broadcast news. Springer (2003)
25. Speranza, M.: Evalita 2007: the named entity recognition task. In: Proceedings of EVALITA (2007)
26. Speranza, M.: The named entity recognition task at Evalita 2009. In: Proceedings of the Workshop Evalita (2009)
27. Srihari, R.K., Niu, C., Li, W., Ding, J.: A case restoration approach to named entity tagging in degraded documents. In: 2013 12th International Conference on Document Analysis and Recognition, vol. 2, pp. 720–720. IEEE Computer Society (2003)
28. Turmo, J., Comas, P., Rosset, S., Galibert, O., Moreau, N., Mostefa, D., Rosso, P., Buscaldi, D.: Overview of QAST 2009-question answering on speech transcriptions. In: CLEF 2009 Workshop (2009)
29. Zanoli, R., Pianta, E.: Intelligenza artificiale—numero speciale su strumenti per l'elaborazione del linguaggio naturale per l'italiano. In: Associazione Italiana per l'Intelligenza Artificiale, vol. **4**, pp. 69–70 (2007)
30. Zanoli, R., Pianta, E., Giuliano, C.: Named entity recognition through redundancy driven classifiers. In: Proceedings of Evalita 9 (2009)

Part II
Tools and Related Methodologies

Semantic Tree Kernels for Statistical Natural Language Learning

Danilo Croce, Roberto Basili and Alessandro Moschitti

Abstract A central topic in Natural Language Processing (NLP) is the design of effective linguistic processors suitable for the target applications. Within this scenario, Convolution Kernels provide a powerful method to directly apply Machine Learning algorithms to complex structures representing linguistic information. The main topic of this work is the definition of the semantically Smoothed Partial Tree Kernel (SPTK), a generalized formulation of one of the most performant Convolution Kernels, i.e. the Tree Kernel (TK), by extending the similarity between tree structures with node similarities. The main characteristic of SPTK is its ability to measure the similarity between syntactic tree structures, which are partially similar and whose nodes can differ but are nevertheless semantically related. One of the most important outcomes is that SPTK allows for embedding external lexical information in the kernel function only through a similarity function among lexical nodes. The SPTK has been evaluated in three complex automatic Semantic Processing tasks: Question Classification in Question Answering, Verb Classification and Semantic Role Labeling. Although these tasks address different problems, state-of-the-art results have been achieved in every evaluation.

Keywords Kernel methods · Tree kernels · Semantic role labeling · Verb classification

D. Croce (✉) · R. Basili
Department of Computer Science, Systems and Production,
University of Roma Tor Vergata, Rome, Italy
e-mail: croce@info.uniroma2.it

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

A. Moschitti
Department of Computer Science and Information Engineering,
University of Trento, Povo (TN), Italy
e-mail: moschitti@disi.unitn.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_5

1 Introduction

Most human knowledge is represented and expressed using language and modern systems in Information Technology need to access the huge amount of information that is stored and constantly produced in the Web. This source of information can be represented in structured form, e.g. stored inside Databases or Data Warehouses, but the vast majority is still produced in an unstructured form, e.g. documents written in natural language. In such a scenario which is also recurrent in real time marketing, semantic web-search, security or exploratory data analysis, the proper application of Natural Language Processing (NLP) techniques allows for more sophisticated access to information, hence providing more natural human-machine interfaces. Traditionally, Information Retrieval (IR) has dealt with representation, storage, organization of, and access to information items, e.g. documents, as described in [1]. However, given the rapid growth of the Web, although people can browse and generate linguistic contents, they still do not provide any effective enrichment of the produced information, e.g. a description of the linguistic content that can be exploited by search engines. The open research questions are: How to exploit this huge source of information? How do we *interpret* this large amount of textual data? Information Retrieval faces nowadays contemporary challenges such as Question Answering (QA) [2] or Sentiment Analysis (SA) [3]: in such tasks, complex and fine-grained linguistic information are involved and a principled model of both linguistic content and background knowledge is needed.

In this scenario, the main goal of Computational Natural Language Learning is to acquire knowledge and models needed to turn texts into meaningful structures (i.e. interpretations). The application of such models provides language learning systems, as largely described in [4, 5]. These allow for generalizing linguistic observations into rules and patterns as statistical models of higher level semantic inferences. Statistical learning methods make the assumption that lexical or grammatical observations are useful hints for modeling different semantic inferences, such as in document topical classification, predicate and role recognition in sentences as well as in question classification in Question Answering. Lexical features here include lemmas, multi-word expressions or Named Entities that can be directly observed in the texts. Features are then generalized into predictive components in the final model, induced from the training examples. A proper model of the linguistic observation is needed as a computational representation. A manual feature encoding, where an expert emphasizes the informative properties with respect to the target problem, represents one solution. This activity produces an artificial representation of the linguistic observations which can be employed by a learning system. One important drawback of such process is the cost of the definition of the proper features for a novel task. Even if the learning algorithm can select the most informative ones, they still need to be defined. Moreover, this activity is very tied to the target application and cannot be easily reused for different tasks. The support for the fast design of accurate automatic systems requires to implicitly derive this information from

the data distribution itself for an automatic engineering of syntactic and semantic properties.

Kernel methods, discussed in [6], have been employed in NLP, as in [7], in order to provide a statistical model able to decouple the problem representation and learning algorithm, still satisfying the above requirements. A kernel function [8], allows us to express the similarity between two objects, that are explanatory of the target problem, without defining their explicit representation and, most importantly, it can be used along with kernel-based learning algorithms, e.g. Support Vector Machines, that represent the state-of-the-art machine learning algorithms applied to NLP tasks. The main idea is that the algorithm can effectively learn the target phenomenon by focusing on the notion of similarity among observations, instead of their representations. A linguistic phenomenon can nevertheless be modeled at a more abstract level making the modeling process easier. For example, which representation would be employed to learn the difference between a correct and incorrect syntactic parse tree? By using the parse tree itself, the learner would focus only on the properties useful for the sake of making a decision. This idea is expanded in Tree Kernels, introduced by [7], that allow to model similarity betweentraining examples as a function of the shared syntactic information, in terms of shared syntactic tree fragments, in the corresponding parses.

In this work, we provide the definition of a semantically **Smoothed Partial Tree Kernel** (**SPTK**) that augments the existing Tree Kernel formulations with node similarity and allows to design effective language learning systems. The underlying idea is to provide a similarity score among lexical nodes depending on the semantic similarity between their labels. SPTK can therefore automatically provide the learning algorithm with a huge set of generalized structural patterns by simply applying the kernel function to the structural representation of the target task instances. Within this scenario, a meaningful similarity measure is thus crucial; in fact the lack of proper lexical generalization is often quoted to bear the main responsibility for significant performance drops in out-of-domain semantic processing tasks, e.g. Semantic Role Labeling, as discussed in [9]. Moreover, due to the expensiveness of developing large scale lexical Knowledge Bases, corpus driven methods will be used to acquire meaning generalizations in an unsupervised fashion, as suggested in [10–12]. A distributional paradigm will enable the extension of the SPTK through the adoption of vector based models of lexical meaning. A large-scale corpus is statistically analyzed and a geometrical space (the Word Space discussed in [11]) is defined: here words are modeled as vectors whose dimensions reflect the words co-occurrence statistics over texts, and the similarity (or distance) among vectors models a notion of semantic similarity between the corresponding words.

A large-scale empirical evaluation of SPTK will be discussed to assess its applicability and robustness. The same kernel will be thus applied to different complex semantic tasks: the *Question Classification* task in a Question Answering setting [13], which represents a sentence classification task; the *Verb Classification* task [14], which is a fundamental topic of computational linguistics research given its importance in understanding the role of verbs in conveying semantics of natural language; the *FrameNet based Semantic Role Labeling* task, which represents a

complex semantic annotation task [15]. In such tasks, the proposed model will not rely on manual feature engineering for linguistic phenomena: the employed discriminative learning algorithm, i.e. Support Vector Machines, will select the most informative features for the target problem without any explicit definition. Furthermore, the lexical information provided by the proposed distributional perspective will be investigated and compared with information obtained from hand-built dictionaries.

In the rest of the paper, Sect. 2 discusses limits of traditional Tree Kernel functions and introduces Distributional Models of Lexical Semantics. Section 3 defines the Smoothed Partial Tree Kernel. Section 4 provides the experimental evaluation. Finally, conclusions are derived in Sect. 5.

2 Tree Kernels and Distributional Models of Lexical Semantics

In order to better understand Tree Kernels and discuss their intrinsic limits, let us describe a task where these kernels have been successfully applied, i.e. Semantic Role Labeling (SRL), as proposed in [15, 16]. Since late 70s, *frame semantics* [17] has been proposed as a model of real world situations or events: a linguistic predicate, called *frame*, is evoked in a sentence through the occurrence of specific *lexical units*, i.e. words (e.g. nouns or verbs) that linguistically express the intended situation. A frame characterizes the set of prototypical semantic roles that describe the participants in the event for all lexical units. SRL is thus the task of automatic recognition of individual predicates together with their main roles, as they are semantically and grammatically realized in input sentences. For example, the following two sentences evoke the STATEMENT frame, i.e. the situation of communicating the act of a SPEAKER or a MEDIUM to address a MESSAGE to some ADDRESSEE using language:

$$[\textit{President Kennedy}]_{\text{SPEAKER}}\ \underline{\text{said}}\ [\textit{to an astronaut}]_{\text{ADDRESSEE}}[\text{“}\textit{Man is still the most extraordinary computer of all.”}]_{\text{MESSAGE}} \quad (1)$$

$$[\textit{The report}]_{\text{MEDIUM}}\ \underline{\text{stated}}\ [\textit{that some problems needed to be solved.}]_{\text{MESSAGE}} \quad (2)$$

The frame is evoked through the lexical units *say* and *state*, and the considered roles are SPEAKER, MEDIUM and MESSAGE. SRL is crucial to support reliable and accurate analysis of unstructured text, in order to enrich it with semantic meta-data and other kinds of information which is implicit in texts.

SRL has been a popular task since the availability of the PropBank [18] and FrameNet [19] annotated corpora and the successful CoNLL evaluation campaigns [20]. In SRL, the role of grammatical information has been outlined since the seminal work by [16], where syntactic parse trees are shown to relate a predicate word to its arguments. State-of-the-art approaches to SRL are based on Support Vector Machines, trained over manually built representations derived from syntactic parse trees (e.g. [9, 21]). As discussed in [22, 23], syntactic information of annotated

examples can be effectively generalized in SRL through the adoption of tree kernel-based learning, without relying on manual feature engineering: as tree kernels model similarity between two training examples as a function of their shared tree fragments, discriminative information is automatically selected by the learning algorithm, e.g., Support Vector Machines.

However, when the availability of training data is limited, the information derived from structural patterns may be not sufficient to discriminate examples. In fact, one important limitation of Tree Kernels is that only string matching between node labels is applied when estimating the number of common substructures. Consequently, this entails a poor lexical generalization. Let us consider the example in sentences 1 and 2. Two phrases like *"President Kennedy said…"* and *"The report stated…"* both evoke the JUDGMENT_COMMUNICATION frame, but the two logical subjects represent two different roles: *President Kennedy* represents a human being, then associated with the SPEAKER role, while *report* is a means of communication, therefore associated with the MEAN role. When a kernel function is applied between the above phrases and *"The mail says…"*, the word *mail* differs both from *president* and *report*, therefore it does not provide any contribution to the overall similarity estimation. Nevertheless, it should be considered that *mail* and *report* are semantically related in the inductive inference process, in order to associate the MEAN role with the above text. On the contrary, the resulting learning algorithm should be provided with all examples where the subject of a verb like *say* is a means of communication in order to learn differences between the SPEAKER and MEAN roles. Problems thus arise when the availability of training data is scant: lexical information should be properly generalized to obtain more informative structural patterns.

A significant research has been done on the study of Distributional Models of lexical semantics to automatically acquire meaningful word generalizations: these models follow the distributional hypothesis [24] and characterize lexical meanings in terms of *context of use* [25]. By inducing geometrical notions of vectors and norms through corpus analysis, they provide a topological definition of semantic similarity, i.e., distance in a space. They can capture the similarity between words such as *report* and *mail*. In supervised language learning, when few examples are available, DMs support cost-effective lexical generalizations, often outperforming knowledge based resources (such as WordNet, as in [26]). Obviously, the choice of the context type determines the type of targeted semantic properties. Wider contexts (e.g., entire documents) are shown to suggest topical relations. Smaller contexts tend to capture more specific semantic aspects, e.g. the syntactic behavior, and better capture paradigmatic relations, such as synonymy. In particular, word space models, as described in [11], define contexts as the words appearing in a n-sized window, centered around a target word. Co-occurrence counts are thus collected in a words-by-words matrix, where each element records the number of times two words co-occur within a single window of word tokens. Moreover, robust weighting schemas are used to smooth counts against too frequent co-occurrence pairs: Pointwise Mutual Information (PMI) scores [27] are commonly adopted. In such statistical paradigm, robust representations can be obtained through intelligent dimensionality reduction methods. According the Latent Semantic Analysis (LSA) technique [28], the original

word-by-word matrix M can be decomposed through Singular Value Decomposition (SVD) [29] into the product of three new matrices: U, S, and V so that S is diagonal and $M = USV^T$. M is then approximated by $M_k = U_k S_k V_k^T$, where only the first k columns of U and V are used, corresponding to the first k greatest singular values. This approximation supplies a way to project a generic term w_i into the k-dimensional space using $W = U_k S_k^{1/2}$, where each row corresponds to the representation vector $\boldsymbol{w_i}$. The original statistical information about M is captured by the new k-dimensional space, which preserves the global structure while removing low-variant dimensions, i.e., distribution noise. Given two words w_1 and w_2, the term similarity function σ is estimated as the cosine similarity between the corresponding projections $\boldsymbol{w_1}$, $\boldsymbol{w_2}$ in the LSA space, i.e. $\sigma(w_1, w_2) = \frac{\boldsymbol{w_1} \cdot \boldsymbol{w_2}}{\|\boldsymbol{w_1}\| \|\boldsymbol{w_2}\|}$. This is known as *Latent Semantic Kernel* (*LSK*), proposed in [30], as it defines a positive semi-definite Gram matrix $\boldsymbol{G} = \sigma(w_1, w_2)\, \forall w_1, w_2$ [8]. σ is thus a valid kernel and can be combined with other kernels, as discussed in the next session.

3 Semantically Smoothed Partial Tree Kernel

The main drawback of pure lexical information is due to its non-compositional nature as the grammatical structure of the sentences is ignored and it is not designed to compute the meanings of phrases. As already addressed in recent works, e.g. [31], the definition of methods able to express the meaning of phrases or sentences as composition operations over geometric representations is a complex problem, and a still largely open issue. Some studies, e.g. [32–36], propose classes of algebraic operators (e.g. tensor products) as effective combination of lexical information. Their focus is to explicitly combine vectors representing words in a phrase in order to obtain a new vector representing the semantics of the entire phrase. These works propose algebraic models of words composition with constraints imposed by the targeted phrase structure. However, these models still work on simple syntactic structures, e.g. they provide a composition between two or three words, although they lack the proper expressivity to be employed in complex tasks.

In this work a different approach is pursued based upon to the idea of convolution kernels: rather than providing an explicit representation of the sentence semantics in terms of word composition, a method is instead defined to estimate the similarity between sentences, embedding this lexical information directly in the similarity function. In this perspective, one interesting approach, proposed in [37], encoded lexical similarity in tree kernels. The model is essentially the Syntactic Tree Kernel (STK), defined in [7], in which syntactic fragments from constituency trees can be matched even if they differ in the leaf nodes (i.e., they are constituted by related words with different surface forms). This kernel has been named *Semantic Syntactic Tree Kernel* (SSTK) and its computation is recursively carried out by the following Δ_{SSTK} function:

- if n_1 and n_2 are not pre-terminals and the productions at n_1 and n_2 are different then

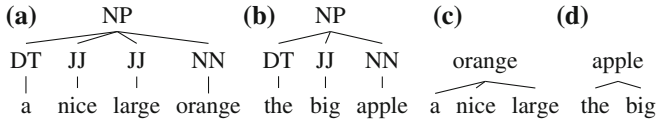

Fig. 1 Examples of syntactic parse trees

$$\Delta_{SSTK}(n_1, n_2) = 0$$

- if n_1 and n_2 are pre-terminals and $label(n_1) = label(n_2)$ then

$$\Delta_{SSTK}(n_1, n_2) = \lambda K_S(ch^1_{n_1}, ch^1_{n_2})$$

- if n_1 and n_2 are not pre-terminals and the productions at n_1 and n_2 are the same[1] then:

$$\Delta_{SSTK}(n_1, n_2) = \lambda \prod_{j=1}^{n_c(n_1)} (1 + \Delta_{SSTK}(ch^j_{n_1}, ch^j_{n_2}))$$

where $label(n_i)$ is the label of node n_i and K_S is a valid term similarity kernel. Note that in constituency parse trees n_1 and n_2 are pre-terminals and they can have only one child (i.e. $ch^1_{n_1}$ and $ch^1_{n_2}$) and such children are words. This kernel uses matching scores between fragments (i.e., features) that depend on the semantic similarity K_S between the corresponding leaves in the syntactic fragments. This allows to match fragments having the same structure but different leaves by assigning a score which is proportional to the product of the lexical similarities of each leaf pair.

Notwithstanding the aforementioned idea is promising and the SSTK provided good results in several NL tasks, such as Question Classification in [37] and Textual Entailment Recognition in [38]. However, the SSTK inherits the intrinsic limitations that reduce the effectiveness of semantic smoothing: in Fig. 1a, b, two simple fragments from a constituency parse tree are shown, representing the two nominal syntagmas "*a nice large orange*" and "*the big apple*", respectively. These short texts are semantically related and a proper lexical similarity could acquire this information by comparing words like *a/the*, *big/large* or *orange/apple*. However, the SSTK does not estimate this similarity among leaves because the production rules `[NP [DT JJ JJ NN]]` and `[NP [DT JJ NN]]` are not the same. Moreover, the SSTK cannot be applied to information represented through dependency parse trees. In Fig. 1c, d, two trees derived from the noun phrases as dependency graphs are shown; it is worth noting that the graph governor is the tree root, while the dependents are the leaves. As the SSTK estimates the K_S only between tree leaves, it is trivial that it cannot be applied to such trees, as their roots are different.

Hereafter, a more general tree kernel is defined and it can be applied to any tree and exploit any combination of lexical similarities thought respecting the syntax enforced

[1] It implies that $n_c(n_1) = n_c(n_2)$.

by the tree. To overcome such issues, the tree kernel proposed in [39], namely the Partial Tree Kernel (PTK), is augmented with node similarity. This allows to use any tree and any lexical similarity metrics between nodes for any position of the tree (not just on the leaves as in [37]). In other words, the new Smoothed PTK (SPTK) can automatically provide the learning algorithm, e.g., Support Vector Machines (SVMs), with a huge set of generalized structural patterns by simply applying it to the structural representation of instances of the target task. Combining lexical and structural kernels provides clear advantages on all-vs-all word similarity, which tends to semantically diverge. Indeed syntax provides the necessary restrictions to compute an effective semantic similarity.

3.1 Smoothed Partial Tree Kernel Definition

As for the evaluation of PTK, the evaluation of the common SPTK rooted in nodes n_1 and n_2 requires the selection of the shared child subsets of the two nodes. Due to the importance of the order of the children, we can use subsequence kernels for their generation. More in detail, let $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ be the set of all possible PT fragment and let the indicator function $I_i(n)$ be equal to 1 if the target f_i is rooted at node n and 0 otherwise, we define the SPTK as:

- If n_1 and n_2 are leaves then $\Delta_{SPTK}(n_1, n_2) = \mu\lambda\sigma_\tau(n_1, n_2)$
- else

$$\Delta_{SPTK}(n_1, n_2) = \mu\sigma_\tau(n_1, n_2) \times \left(\lambda^2 + \sum_{\boldsymbol{I}_1, \boldsymbol{I}_2, l(\boldsymbol{I}_1)=l(\boldsymbol{I}_2)} \lambda^{d(\boldsymbol{I}_1)+d(\boldsymbol{I}_2)} \prod_{j=1}^{l(\boldsymbol{I}_1)} \Delta_{SPTK}(c_{n_1}(\boldsymbol{I}_{1j}), c_{n_2}(\boldsymbol{I}_{2j}))\right) \quad (3)$$

Here the formulation is similar to the PTK, c_{n_1} and c_{n_2} are the ordered child sequences of n_1 and n_2 respectively, while $I_1 = \langle I_{11}, I_{12}, I_{13}, \ldots\rangle$ and $I_2 = \langle I_{21}, I_{22}, I_{23}, \ldots\rangle$ are index sequences associated with the ordered child sequences such that $c_{n_1}(\boldsymbol{I}_{1j})$ and $c_{n_2}(\boldsymbol{I}_{2j})$ are the jth children in the two sequences respectively. The function $l(\cdot)$ returns the sequence length. As for PTK, two decay factors are employed: $0 < \mu \leq 1$ for the height of the tree and $0 < \lambda \leq 1$ for the length of the child sequences. It follows that both larger trees and subtrees built on child subsequences that contain gaps are penalized depending on the exponent $d(\boldsymbol{I}_1) = \boldsymbol{I}_{1l(\boldsymbol{I}_1)} - \boldsymbol{I}_{11}$ and $d(\boldsymbol{I}_2) = \boldsymbol{I}_{2l(\boldsymbol{I}_2)} - \boldsymbol{I}_{21}$, i.e. the width of the production rule.

The novelty of SPTK is represented by the embedding of a similarity function σ_τ between nodes which are typed according to τ. It is more general than the SSTK as it depends on the position of the node pairs within the trees, i.e. non terminals nodes and leaves. Furthermore, the overall SPTK is neutral with respect to the target linguistic problems discussed in this work. Obviously, the similarity function between nodes must be carefully designed in order to grant effectiveness in the target semantic processing task: in fact, the SPTK would enumerate and compare any possible node

Algorithm 1 $\sigma_\tau(n_1, n_2, lw)$

$\sigma_\tau \leftarrow 0$,
if $\tau(n_1) = \tau(n_2) = \text{SYNT} \wedge label(n_1) = label(n_2)$ **then**
  $\sigma_\tau \leftarrow 1$
end if
if $\tau(n_1) = \tau(n_2) = \text{POS} \wedge label(n_1) = label(n_2)$ **then**
  $\sigma_\tau \leftarrow 1$
end if
if $\tau(n_1) = \tau(n_2) = \text{LEX} \wedge pos(n_1) = pos(n_2)$ **then**
  $\sigma_\tau \leftarrow \sigma_{LEX}(n_1, n_2) \times lw$
end if
return σ_τ

pairs, including non terminal nodes. From a linguistic perspective this is problematic as each node reflects a specific aspect of data and the comparison between nodes of different nature, e.g. syntactic nodes like `NP` or `VP`, and lexical nodes like `apple` or `orange` should be avoided. The similarity function $\sigma_\tau(n_1, n_1)$ between two nodes n_1 and n_2 must depend on the nodes' type τ. An example of σ_τ is shown by Algorithm 1: given two nodes n_1 and n_2, it applies a different similarity for each node type. Types are described by τ and are divided into: syntactic categories (i.e., $\tau = \text{SYNT}$), POS-Tag labels (i.e., $\tau = \text{POS}$) or a lexical (i.e., $\tau = \text{LEX}$) type. In this example we require a hard match between non lexical nodes, i.e. assigning 0/1 similarity for SYNT and POS nodes. For LEX type, a lexical kernel K_{LEX}, introduced in Sect. 2, is applied between words sharing the same POS-Tag. It means that words that belong to different shallow grammatical classes are never considered compatible, e.g., nouns with a verbs or adjectives.

The lexical similarity function is therefore crucial in order to provide a meaningful kernel estimation. As discussed in the following sections when focusing on empirical evaluations, this lexical kernel can be acquired from an existing lexicon or directly through Distributional modeling. Indeed, such general formulation also allows for using weighting schemes with different similarity functions. For examples, in Algorithm 1 the contribution of the lexical information is amplified (or reduced) trough a *lexical weight* (lw), that multiplies the similarity function between lexemes.

The underlying principle that allows employing SPTK in a kernel based learning algorithms, e.g. Support Vector Machine, is that SPTK must be a valid kernel. In order to demonstrate its validity, let us consider the node similarity function σ as a string matching between node labels and $\lambda = \mu = 1$. Each recursive step of Eq. 3 can be seen as a summation of $(1 + \prod_{j=1}^{l(\boldsymbol{I}_1)} \Delta_{STK}(c_{n_1}(\boldsymbol{I}_{1j}), c_{n_2}(\boldsymbol{I}_{2j})))$, i.e. the Δ_{STK} recursive equation, for all subsequences of children $c_{n_1}(\boldsymbol{I}_{1j})$. In other words, PTK is a summation of an exponential number of STKs, which are valid kernels. It follows that PTK is a kernel. Note that the multiplication by λ and μ elevated to any power only depends on the target fragment. Thus, it just gives an additional weight to the fragment and does not violate the Mercer's condition, that is discussed in [6]. In contrast, the multiplication by $\sigma(n_1, n_2)$ does depend on both comparing

examples, i.e. on n_1 and n_2. However, if the matrix $\left[\sigma(n_1, n_2)\right]\forall n_1, n_2 \in f \in \mathcal{F}$ is positive semi-definite, a decomposition exists such that $\sigma(n_1, n_2) = \phi(n_1)\phi(n_2) \Rightarrow \Delta_\sigma(n_1, n_2)$ can be written as $\sum_{i=1}^{|\mathcal{F}|} \phi(n_1)\chi_i(n_1)\phi(n_2)\chi_i(n_2) = \sum_{i=1}^{|\mathcal{F}|} \phi_\sigma(n_1)\phi_\sigma(n_2)$, which proves SPTK to be a valid kernel.

3.2 Proposed Computational Structures

The feature space generated by the structural kernels, presented in the previous section, obviously depends on the input structures. In case of PTK and SPTK different tree representations may lead to engineer more or less effective syntactic/semantic feature spaces, as discussed in [7, 39]. Due to their nature, *constituency parse* trees can be easily employed in the TK estimation. Given the following sentence:

(s1) *What is the width of a football field?*

The representation tree for a phrase structure paradigm leaves little room for variations as shown by the constituency tree (CT) in Fig. 2. We apply lemmatization to the lexemes to improve generalization and, at the same time, we add to them a generalized PoS-tag, i.e. noun (n::), verb (v::), adjective (::a), determiner (::d) and so on. This is useful in forcing similarity to insist only between lexemes of the same grammatical category.

In contrast, the conversion of *dependency structures* in computationally effective trees (for the above kernels) is not straightforward. We need to define the role of lexemes, PoS-tags and grammatical functions (GR). In order to transform the dependency graph in a tree structure, the edge label can be associated with tree nodes to surrogate the syntactic information. The basic idea of our structures is to use (i) one of the three kinds of information above the central nodes, from which dependencies are drawn and (ii) all the other information as features (in terms of dominated nodes) attached to the formed ones.

We define three main versions to represent dependency trees, such as the one shown in Fig. 3:

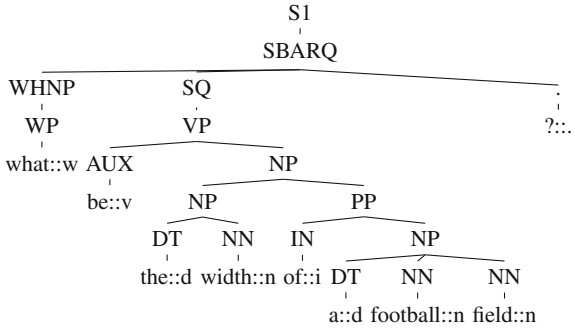

Fig. 2 Constituent Tree (CT)

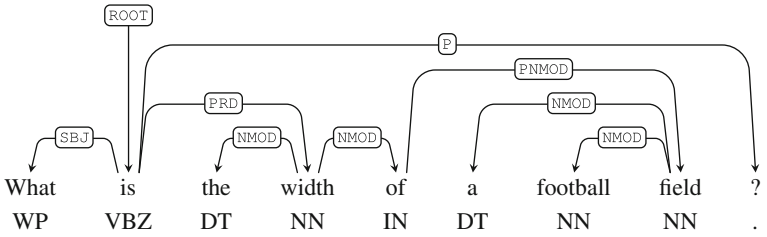

Fig. 3 Dependency Parse Tree

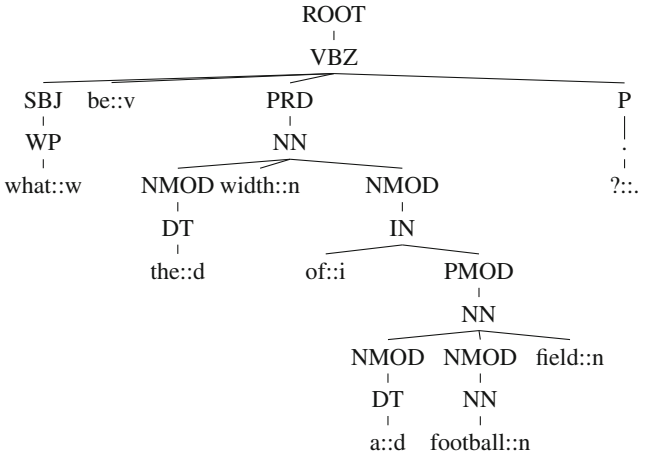

Fig. 4 PoS-Tag Centered Tree (PCT)

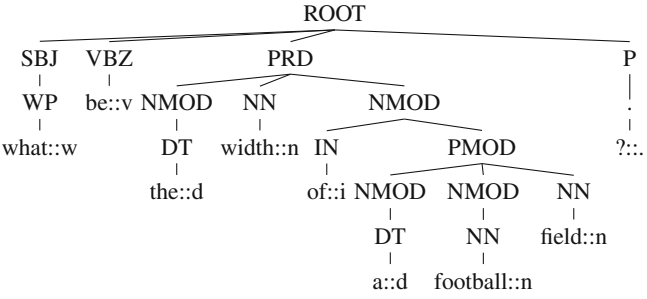

Fig. 5 Grammatical Relation Centered Tree (GRCT)

- the PoS-Tag Centered Tree (PCT), e.g. see Fig. 4, where the GR is added as father and the lexical as a child;
- the GR Centered Tree (GRCT), e.g. see Fig. 5, where the PoS-Tags are children of GR nodes and fathers of their associated lexemes;

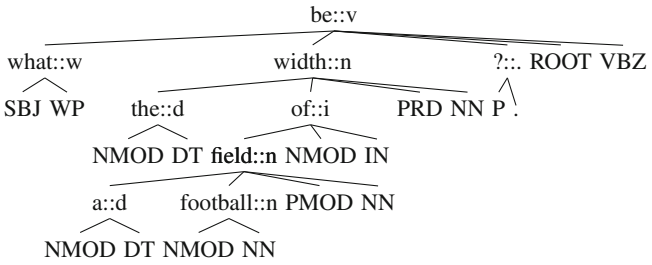

Fig. 6 Lexical Centered Tree (LCT)

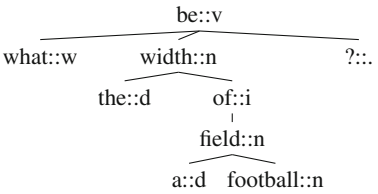

Fig. 7 Lexical Only Centered Tree (LOCT)

- the Lexical Centered Tree (LCT), e.g. see Fig. 6, in which both GR and PoS-Tag are added as the rightmost children.

To better study the role of the above dependency structures, especially from a performance perspective, we specify additional structures. Figure 7 shows the Lexical Only Centered Tree (LOCT) which is directly derived from the parse tree. It only accounts on the lexemes, where untyped binary relations are used for recursive structures. The grammatical generalization provided by the syntactic edge labels is thus neglected. In order to have a meaningful comparison, two trees whose structures does not reflect the sentence syntactic information are here defined. Figure 8 shows the Lexical and PoS-Tag Sequences Tree (LPST) in the form of a flattened tree with two levels, one for PoS-Tag information, where lexemes are simply added as leaves. Finally, in Fig. 9 only lexical items are leaves of a single root node. These

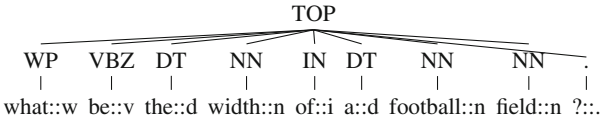

Fig. 8 Lexical and PoS-Tag Sequences Tree (LPST)

Fig. 9 Lexical Sequences Tree (LST)

two structures are interesting as they allow to employ a PTK or SPTK to surrogate the Sequence Kernel [40].

4 Experimental Evaluation

A large scale empirical evaluation is here discussed to describe the application of SPTK to a different semantic processing task: the Question Classification task in Sect. 4.2, the Verb Classification task in Sect. 4.3 and the Semantic Role Labeling task in Sect. 4.4. The aim of the following experiments is to analyze different levels of representation, i.e. structure, for syntactic dependency parses. Most importantly, the role of lexical similarity embedded in syntactic structures will be investigated.

4.1 General Experimental Setup

The following semantic processing task are modeled as a classification problem, where a SVM classifier is employed. For SVM learning, we extended the SVM-LightTK software[2] (which includes structural kernels in SVMLight [41]) with the smooth match between tree nodes. For generating constituency trees, we used the Charniak parser [42] whereas we applied LTH syntactic parser (described in [43]) to generate dependency trees. Lexical similarity is derived through the distributional analysis of UkWaC [44], which is a large scale document collection made by 2 billion tokens. More specifically, to build the matrix M, POS tagging is first applied so that its rows are pairs ⟨lemma, ::POS⟩, or lemma::POS in brief. The contexts of such items are the columns of M and are short windows of size $[-3, +3]$, centered on the items. This allows for better capturing syntactic properties of words. The most frequent 20,000 items are selected along with their 20 k contexts. The entries of M are the point-wise mutual information between them. The SVD reduction is then applied to M, with a dimensionality cut of $l = 250$. In Question Classification experiments the contribution of distributional models is compared with a resource based similarity derived from the word list (WL) provided in [13].

SVM-LightTK is applied to the different tree representations discussed in Sect. 3.2. We experiment with multi-classification, which we model through *one-vs-all* scheme

[2] http://disi.unitn.it/moschitti/Tree-Kernel.htm.

Table 1 Accuracy of several structural kernels on different structures for coarse and fine grained QC

	COARSE						FINE					
	NO		LSA		WL		NO		LSA		WL	
	lw	Acc. (%)	*lw*	Acc. (%)	*lw*	Acc. (%)	*lw*	Acc. (%)	*lw*	Acc. (%)	*lw*	Acc. (%)
CT	4	90.80	2	91.00	5	92.20	4	84.00	5	83.00	7	86.60
GRCT	3	**91.60**	4	92.60	2	**94.20**	3	83.80	4	83.20	2	85.00
LCT	1	90.80	1	**94.80**	1	**94.20**	0.33	**85.40**	1	86.20	0.33	**87.40**
LOCT	1	89.20	1	93.20	1	91.80	1	**85.40**	1	**86.80**	1	87.00
LST	1	88.20	1	85.80	1	89.60	1	84.00	1	80.00	1	85.00
LPST	3	89.40	1	89.60	1	92.40	3	84.20	4	82.20	1	84.60
PCT	4	91.20	4	92.20	5	93.40	4	84.80	5	84.00	5	85.20
CT-STK	–	91.20	–	–	–	–	–	82.20	–	–	–	–
BOW	–	88.80	–	–	–	–	–	83.20	–	–	–	–

by selecting the category associated with the maximum SVM margin. The quality of such classification is measured with accuracy. We determine the statistical signicance by using the model described in [45] and implemented in [46].

The parameterization of each classifier is carried on a held-out set and concerns with the setting of the trade-off parameter (option -c) and the Leaf Weight (lw) (see Algorithm 1), which is used to linearly scale the contribution of the leaf nodes. In contrast, the cost-factor parameter of the SVM-LightTK is set as the ratio between the number of negative and positive examples for attempting to have a balanced Precision/Recall.

4.2 Question Classification

The typical architecture of a QA system includes three main phases: question processing, document retrieval and answer extraction [2]. Question processing is usually centered around the so called Question Classification task. It maps a question into one of k predefined answer classes, thus posing constraints on the search space of possible answers. For these experiments, we used the UIUC dataset [13]. It is composed by a training set of 5,452 questions and a test set of 500 questions.[3] Question classes are organized in two levels: 6 coarse-grained classes (like `ENTITY` or `HUMAN`) and 50 fine-grained sub-classes (e.g. `Plant`, `Food` as subclasses of `ENTITY`).

The outcome of the several kernels applied to several structures for the coarse and fine grained QC is reported in Table 1. Since PTK and SPTK are typically used in our experiments, to have a more compact acronym for each model, we associate the

[3] http://cogcomp.cs.illinois.edu/Data/QA/QC/.

latter with the name of the structure, i.e. this indicates that PTK is applied to it. Then the presence of the subscript $_{WL}$ and $_{LSA}$ indicates that SPTK is applied along with the corresponding similarity, e.g. LCT$_{WL}$ is the SPTK kernel applied to LCT structure, using WL similarity. The first column shows the experimented models, obtained by applying PTK/SPTK to the structures described in Sect. 3.2. The last two rows are: CT-STK, i.e. Syntactic Tree Kernel, proposed in [7] applied to a constituency tree and BOW, which is a linear kernel applied to lexical vectors. Column 2, 3 and 4 report the accuracy using no, LSA and WL similarity, where lw is the amplifying parameter, i.e. weight, associated with the leaves in the tree. The last three columns refer to the fine- grained task.

It is worth nothing that when no similarity is applied: (i) BOW produces high accuracy, i.e. 88.8 % but it is improved by STK, current state-of-the-art[4] in QC; (ii) PTK applied to the same tree of STK produces a slightly lower value (non-statistically significant difference); (iii) interestingly, when PTK is instead applied to dependency structures, it improves STK, i.e. 91.60 versus 91.40 % (although not significantly); and (iv) LCT, strongly based on lexical nodes, is the less accurate, i.e 90.80 % since it is obviously subject to data sparseness (fragments only composed by lexicals are very sparse). The very important results can be noted when lexical similarity is used, i.e. SPTK is applied: (a) all the syntactic-base structures using both LSA or WL improve the classification accuracy (b) CT gets the lowest improvement whereas LCT achieves an impressive result of 94.80 %, i.e. more than 41 % of relative error reduction. It seems that the lexical similar paths when driven by syntax produces accurate features. Indeed, when syntax is missing such as for the unstructured lexical path of LST$_{LSA}$, the accuracy does not highly improve or may also decrease. Additionally, the result of our best model is so high that its errors only refer to questions like *What did Jesse Jackson organize?*, where the classifier selected `Entity` instead of `Human` category. These are clear examples where a huge amount of background knowledge is needed. Finally, on the fine grained experiments LCT still produces the most accurate outcome again exceeding the state-of-the-art [47], where WL significantly improves on all models (CT included).

4.3 Verb Classification

Verb classification is a fundamental topic of computational linguistics research given its importance for understanding the role of verbs in conveying semantics of natural language (NL). Currently, a lot of interest has been devoted to the VerbNet verb categorization scheme [48]. However, the definition of models for optimally combining lexical and syntactic constraints is still far for being accomplished. In particular, the exhaustive design and experimentation of lexical and syntactic features for learning

[4] Note that in [37], higher accuracy values for smoothed STK are shown for different parameters but the best according to a validation set is not highlighted.

verb classification appears to be computationally problematic. For example, the verb **order** can belongs to the two VerbNet classes:

- The class 60.1, i.e., *order someone to do something* as shown in: *The Illinois Supreme Court **ordered** the commission to audit Commonwealth Edison's construction expenses and refund any unreasonable expenses.*
- The class 13.5.1: *order or request something* like in: *... Michelle blabs about it to a sandwich man while **ordering** lunch over the phone.*

Clearly, the syntactic realization can be used to discern the cases above but it would not be enough to correctly classify the following verb occurrence: "*... ordered the lunch to be delivered ...*" in Verb class 13.5.1. For such a case, selectional restrictions are needed.

The implicit feature space generated by structural kernels and the corresponding notion of similarity between verbs obviously depend on the input structures. First we employed the constituency tree (CT) representation, enriching the target verb node with the `target` label. Here, we apply tree pruning to reduce the computational complexity of tree kernels as it is proportional to the number of nodes in the input trees. Accordingly, we only keep the subtree dominated by the target `VP` by pruning from it all the `S`-nodes along with their subtrees (i.e., all nested sentences are removed). To encode dependency structure information in a tree we employed the GR Centered Tree (GRCT) and the Lexical Centered Tree (LCT); for both trees, the pruning strategy only preserves the verb node, its direct ancestors (father and siblings) and its descendants up to two levels (i.e., direct children and grandchildren of the verb node). Note that our dependency tree can capture the semantic head of the verbal argument along with the main syntactic construct, e.g., *to audit*.

In these experiments, we tested the impact of our different verb representations using different kernels, similarities and parameters. We also compared with simple bag-of-words (BOW) models and the state-of-the-art. In particular, we used the same verb classification setting of [14]: sentences are drawn from the Semlink corpus [49], which consists of the PropBanked Penn Treebank portions of the Wall Street Journal. It contains 113 K verb instances, 97 K of which are verbs represented in at least one VerbNet class. Semlink includes 495 verbs, whose instances are labeled with more than one class (including one single VerbNet class or none). We used all instances of the corpus for a total of 45,584 instances for 180 verb classes. When instances labeled with the *none* class are not included, the number of examples becomes 23,719. We used 70 % of instances for training and 30 % for testing.

Our verb (multi) classifier is designed with the *one-vs-all* [50] multi-classification schema. This uses a set of binary SVM classifiers, one for each verb class (frame) i. The sentences whose verb is labeled with the class i are positive examples for the classifier i. The sentences whose verbs are compatible with the class i but evoking a different class or labeled with *none* (no current verb class applies) are added as negative examples. In the classification phase the binary classifiers are applied by (i) only considering classes that are compatible with the target verbs; and (ii) selecting the class associated with the maximum positive SVM margin. If all classifiers provide a negative score the example is labeled with *none*. To assess the performance of

our settings, we also derive a simple baseline based on the bag-of-words (BOW) model. For it, we represent an instance of a verb in a sentence using all words of the sentence (by creating a special feature for the predicate word). We also used a Sequence Kernel (SK) applied to the LST structure, described in Sect. 3.2; for efficiency reasons,[5] we only consider the 10 words before and after the predicate with subsequence features of length up to 5. Table 2 reports the accuracy of different models for VerbNet classification. It should be noted that: first, LST produces a much higher accuracy than BOW, i.e., 82.08 versus 79.08 %. On one hand, this is generally in contrast with standard text categorization tasks, for which n-gram models show accuracy comparable to the simpler BOW. On the other hand, it simply confirms that verb classification requires the dependency information between words (i.e., at least the sequential structure information provided by LST). Second, LST is 2.56 % points below the state-of-the-art achieved in [14] (BR), i.e., 82.08 versus 84.64. In contrast, STK applied to our representation (CT, GRCT and LCT) produces comparable accuracy, e.g., 84.83, confirming that syntactic representation is needed to reach the state-of-the-art. Third, PTK, which produces more general structures, improves over BR by almost 1.5 (statistically significant result) when using our dependency structures GRCT and LCT. CT does not produce the same improvement since it does not allow PTK to directly compare the lexical structure (lexemes are all leaf nodes in CT and to connect some pairs of them very large trees are needed). Finally, the best model of SPTK (i.e., using LCT) improves over the best PTK (i.e., using LCT) by almost 1 point (statistically significant result): this difference is only given by lexical similarity. SPTK improves on the state-of-the-art by about 2.08 absolute percent points, which, given the high accuracy of the baseline, corresponds to 13.5 % of relative error reduction.

Table 2 VerbNet accuracy with the *none* class

	STK		PTK		SPTK	
	lw	Acc. (%)	*lw*	Acc. (%)	*lw*	Acc. (%)
CT	–	83.83	8	**84.57**	8	84.46
GRCT	–	84.83	8	85.15	8	**85.28**
LCT	–	77.73	0.1	86.03	0.2	**86.72**
Br. et Al	84.64 %					
BOW	79.08 %					
LST	82.08 %					

[5] The average running time of the SK is much higher than the one of PTK. When a tree is composed by only one level PTK collapses to SK.

4.4 FrameNet Role Classification

To verify that our findings are general and that our syntactic/semantic dependency kernels can be effectively exploited for diverse NLP tasks, we experimented with a completely different application, i.e. FrameNet SRL classification. Given a predicate (the lexical unit, as described in Sect. 2) and a set of arguments, the Role Classification consists in the assignment of the proper role label to each argument. We used the FrameNet version 1.3 with the 90/10 % split between training and test set (i.e. 271,560 and 30,173 examples respectively), as defined in [9], one of the best system for FrameNet parsing. We used the LTH dependency parser. LSA was applied to the BNC corpus, the source of the FrameNet annotations.

For each of 648 frames, we applied SVM along with the best models for QC, i.e. GRCT and LCT, to learn its associated binary role classifiers (RC) for a total of 4,254 classifiers. For example, Fig. 10 shows the LCT representation of the first two roles of the following sentence:

$$[\textit{Bootleggers}]_{\text{CREATOR}}, \textit{then } \textbf{copy } [\textit{the film}]_{\text{ORIGINAL}}$$
$$[\textit{onto hundreds of VHS tapes}]_{\text{GOAL}}$$

Table 3 shows the results of the different multi-classifiers. GRCT and LCT show a large accuracy, i.e. 87.60 %. This improves up to 88.74 % by activating the LSA similarity. The combination GRCT_{LSA}+LCT_{LSA} significantly improves the above model, achieving 88.91 %. This is very close to the state-of-the-art of SRL for classification (using a single classifier, i.e. no joint model), i.e. 89.6 %, achieved in [9]. These results thus confirm the idea that a lexical generalization allows to improve the quality of the Argument Classification, especially for examples where the syntactic information alone is not discriminative, like the examples of Sentences 1 and 2. Finally, it should be noted that, to learn and test the SELF_MOTION multi-classifier, containing 14,584 examples, distributed on 22 roles, SVM-SPTK employed 1.5 h and 10 min, respectively.[6]

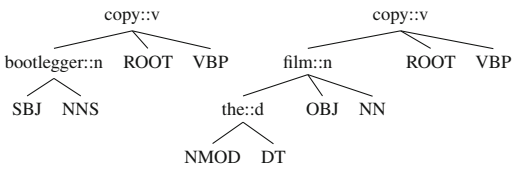

Fig. 10 LCT Examples for argument roles

[6] Using one of the 8 processors of an Intel(R) Xeon(R) CPU E5430 @ 2.66 GHz machine, 32 Gb Ram.

Table 3 Argument Classification Accuracy

Kernel	Accuracy (%)
GRCT	87.60
$GRCT_{LSA}$	88.61
LCT	87.61
LCT_{LSA}	88.74
GRCT + LCT	87.99
$GRCT_{LSA}$ + LCT_{LSA}	**88.91**

5 Conclusions

This paper has proposed a study on representation of dependency structures for the design of effective structural kernels. Most importantly, we have defined a new class of kernel functions, i.e. SPTK, that carry out syntactic and lexical similarity on the above structures. This allows for automatically generating feature spaces of generalized syntactic/semantic dependency substructures. To test our models, we carried out experiments on Question Classification, Verb Classification and Semantic Role Labeling. These show that by exploiting the similarity between two sets of words carried out according to their dependency structure leads to an unprecedented result, whereas no structure is used the accuracy does not significantly improves. We have also provided a fast algorithm for the computation of SPTK and empirically shown that it can easily scale. Such result enables many promising future research directions: the most important being the use of SPTK for many NLP tasks with many different similarities.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Boston (1999)
2. Kwok, C.C., Etzioni, O., Weld, D.S.: Scaling question answering to the web. In: World Wide Web, pp. 150–161 (2001)
3. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
4. Jelinek, F.: Statistical Methods for Speech Recognition. The MIT Press, Cambridge (1998)
5. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
6. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience, New York (1998)
7. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS'2001), pp. 625–632 (2001)
8. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004).
9. Johansson, R., Nugues, P.: The effect of syntactic representation on semantic role labeling. In: Proceedings of COLING, Manchester, 18–22 Aug 2008
10. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. Comput. Linguist. **33**(2), (2007)

11. Sahlgren, M.: The Word-space model. PhD thesis, Stockholm University (2006)
12. Schütze, H.: Word space. In: Advances in Neural Information Processing Systems 5, pp. 895–902. Morgan Kaufmann (1993)
13. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of ACL'02 (2002)
14. Brown, S.W., Dligach, D., Palmer, M.: Verbnet class assignment as a WSD task. In: Proceedings of the Ninth International Conference on Computational Semantics, IWCS'11, pp. 85–94. Association for Computational Linguistics, Stroudsburg (2011)
15. Gildea, D., Palmer, M.: The necessity of parsing for predicate argument recognition. In: Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02), Philadelphia (2002)
16. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Comput. Linguist. **28**(3), 245–288 (2002)
17. Fillmore, C.J.: Frames and the semantics of understanding. Quaderni di Semantica **4**(2), 222–254 (1985)
18. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: an annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2005)
19. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of COLING-ACL, Montreal, Canada (1998)
20. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In: Proceedings of CoNLL-2005, Ann Arbor, Michigan, June 2005, pp. 152–164
21. Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J.H.: Support vector learning for semantic argument classification. Mach. Learn. J. **60**(1–3), 11–39 (2005)
22. Coppola, B., Moschitti, A., Riccardi, G.: Shallow semantic parsing for spoken language understanding. In: Proceedings of NAACL'09, pp. 85–88. Morristown, NJ (2009)
23. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Comput. Linguist. **34**(2), 193–224 (2008)
24. Firth, J.: A synopsis of linguistic theory 1930–1955. In: Studies in Linguistic Analysis. Philological Society, Oxford (1957) reprinted in Palmer, F. (ed.) Selected Papers of J. R. Firth, Longman, Harlow (1968)
25. Wittgenstein, L.: Philosophical Investigations. Blackwells, Oxford (1953)
26. Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., Hovy, E.: ISP: Learning inferential selectional preferences. In: Proceedings of HLT/NAACL (2007)
27. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010)
28. Landauer, T., Dumais, S.: A solution to plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychol. Rev. **104** (1997)
29. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. J. Soc. Ind. Appl. Math.: Ser. B, Numer. Anal. **2**(2), 205–224 (1965)
30. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. In: Brodley, C., Danyluk, A. (eds.) Proceedings of the 18th International Conference on Machine Learning (ICML-01), pp. 66–73. Morgan Kaufmann Publishers, San Francisco, Williams College (2001)
31. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cogn. Sci. **34**, 1388–1429 (2010)
32. Annesi, P., Storch, V., Basili, R.: Space projections as distributional models for semantic composition. In: Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing'12. Springer (2012)
33. Baroni, M., Lenci, A.: One distributional memory, many semantic spaces. In: Proceedings of the GEMS 2009 Workshop. GEMS'09, pp. 1–8. Stroudsburg (2009)
34. Clark, S., Pulman, S.: Combining symbolic and distributional models of meaning. In: Proceedings of the AAAI Spring Symposium on Quantum Interaction, pp. 52–55 (2007)
35. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. In: Proceedings of EMNLP 2011, Edinburgh, Scotland, UK (2011)

36. Zanzotto, F.M., Korkontzelos, I., Fallucchi, F., Manandhar, S.: Estimating linear models for compositional distributional semantics. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10), pp. 1263–1271. Association for Computational Linguistics, Stroudsburg (2010)
37. Bloehdorn, S., Moschitti, A.: Structure and semantics for expressive text kernels. In: Proceedings of CIKM (2007)
38. Mehdad, Y., Moschitti, A., Zanzotto, F.M.: Syntactic/semantic structures for textual entailment recognition. In: HLT-NAACL, pp. 1020–1028 (2010)
39. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: 17th European Conference on Machine Learning, Proceedings, Machine Learning: ECML 2006, pp. 318–329. ECML, Berlin, Germany, Sept 2006
40. Cancedda, N., Gaussier, E., Goutte, C., Renders, J.M.: Word sequence kernels. J. Mach. Learn. Res. **3**, 1059–1082 (2003)
41. Joachims, T.: Estimating the generalization performance of a SVM efficiently. In: Proceedings of ICML'00 (2000)
42. Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of NAACL'00 (2000)
43. Johansson, R., Nugues, P.: Dependency-based syntactic-semantic analysis with PropBank and NomBank. In: Proceedings of the Twelfth Conference on Natural Language Learning (CoNLL 2008), pp. 183–187. Manchester (2008)
44. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: a collection of very large linguistically processed web-crawled corpora. Lang. Res. Eval. **43**(3), 209–226 (2009)
45. Yeh, A.S.: More accurate tests for the statistical significance of result differences. In: COLING, pp. 947–953 (2000)
46. Padó, S.: User's guide to `sigf`: significance testing by approximate randomisation (2006)
47. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 26–32. ACM Press (2003)
48. Schuler, K.K.: VerbNet: A broad-coverage, comprehensive verb lexicon. PhD thesis, University of Pennsylyania (2005)
49. Loper, E., ting Yi, S., Palmer, M.: Combining lexical resources: mapping between propbank and verbnet. In: Proceedings of the 7th International Workshop on Computational Linguistics (2007)
50. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. **5**, 101–141 (2004)

Distributional Models for Lexical Semantics: An Investigation of Different Representations for Natural Language Learning

Danilo Croce, Simone Filice and Roberto Basili

Abstract Language learning systems usually generalize linguistic observations into rules and patterns that are statistical models of higher level semantic inferences. When the availability of training data is scarce, lexical information can be limited by data sparseness effects and generalization is thus needed. Distributional models represent lexical semantic information in terms of the basic co-occurrences between words in large-scale text collections. As recent works already address, the definition of proper distributional models as well as methods able to express the meaning of phrases or sentences as operations on lexical representations is a complex problem, and a still largely open issue. In this paper, a perspective centered on Convolution Kernels is discussed and the formulation of a Partial Tree Kernel that integrates syntactic information and lexical generalization is studied. Moreover a large scale investigation of different representation spaces, each capturing a different linguistic relation, is provided.

Keywords Distributional lexical semantics · Kernel methods · Question classification

1 Introduction

Language learning systems usually generalize linguistic observations into rules and patterns that are statistical models of higher level semantic inferences. Statistical learning methods make the assumption that lexical or grammatical observations are useful hints for modeling different semantic inferences, such as in document

D. Croce (✉) · S. Filice · R. Basili
Department of Computer Science, Systems and Production,
University of Roma Tor Vergata, Via Del Politecnico 1, 00133 Rome, Italy
e-mail: croce@info.uniroma2.it

S. Filice
e-mail: filice@info.uniroma2.it

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

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_6

topical classification, predicate and role recognition in sentences, as well as question classification in Question Answering. Features are then generalized into predictive components in the final model that is effectively induced from the training examples. When the availability of training data is scarce, lexical information (such as lemmas, multiword expressions or Named Entities) can be limited by data sparseness effects and generalization is thus needed. Suitable representations of word meaning as derived from texts play a crucial role here, being a core problem in Computational Linguistics.

Geometrical models represent lexical semantic information through the analysis of observations across large-scale corpora. The core idea is that the meaning of a word can be described by the set of textual contexts in which it appears (*Distributional Hypothesis* as described in [1]). Words can be represented as vectors whose components reflect the corresponding contexts: two words close in the space (i.e. they have similar contexts) are likely to be related by some type of generic semantic relation, either paradigmatic (e.g. synonymy, hyperonymy, antonymy) or syntagmatic (e.g. meronymy, conceptual and phrasal association), as observed in [2]. Semantic spaces have been widely used for representing the meaning of words or other lexical entities [3], with successful applications in lexical disambiguation [4], harvesting thesauri [5], Name Entity Classification [6] or the Semantic Role Labeling task, as in [7].

Obviously, lexical information usually implies different words to provide different contributions but usually neglects other crucial linguistic properties, such as word ordering. In some approaches, symbolic expressions, i.e. pseudo-words, are extracted from the syntactic parse trees and used as lexical features. However, the overall limitation of geometrical models is their non-compositional nature. In general, they ignore the grammatical structure of sentences. On the other side, even in more structured versions, they do not integrate any syntax in computing the meanings of phrases, as that they implicitly do for words, e.g. [8]. As recent works have already addressed, e.g. [9], the definition of methods able to express the meaning of phrases, or sentences, through composition operations acting over the underlying lexical representations, is a complex problem, and a still largely open issue. Some studies, e.g. [10–12], propose classes of algebraic operators (e.g. tensor products) to effectively combine the lexical information of constituents. Their focus is to explicitly combine vectors representing words of a phrase in order to obtain a new vector that represents the semantics of the entire phrase.

In this work we follow a different approach inspired by *Convolution Kernel* methods introduced in [13]. The idea is that we do not need to compose the geometric representation of words to estimate the similarity among two sentences, but instead we compute it in an implicit space by exploiting their grammatical structure. Such kernel methods are very useful as they can be applied to many well-known learning algorithms, such as Perceptrons or Support Vector Machines (SMVs). A key property of these algorithms is that the only operation they require is the evaluation of dot products between pairs of examples. The dot product can be replaced with a Mercer kernel, implicitly mapping feature vectors into a much larger feature space where the original algorithm can be applied and the most representative features can be auto-

matically selected. Automatic feature engineering of syntactic or shallow semantic structures has been carried out by means of Syntactic Tree Kernels (STK), e.g. [14].

One main limitation of these approaches is that they apply a hard matching between node labels: two words, e.g. *boy* and *child*, are different and will provide no contribution to the overall similarity estimation, although they support the same inductive inferences in a learning process. A more effective similarity estimation between tree structures should consider lexical generalization and should apply a more expressive strategy than a simple string matching between labels. Most notably, the work in [15] encodes lexical similarity in tree kernels. This is essentially the STK in which syntactic fragments from constituency trees can be matched, even if they only differ in the leaf nodes (i.e. they have different surface forms). This implies matching scores lower than one, depending on the semantic similarity of the corresponding leaves in the syntactic fragments. In [16] a more general formulation of a semantically *Smoothed Partial Tree Kernel* (SPTK) has been provided. With respect to [15] it can be applied to every tree node (not only the leaves) and it has been successfully applied to dependency parse trees.

One open issue is that different kinds of generalizations can be obtained by changing the adopted lexical similarity function, as this generalizes different semantic aspects of the involved words. While similarity can be modeled directly over lexical resources, e.g. WordNet as discussed in [17], their development can be very expensive thus limiting the coverage of the resulting convolution kernel, especially in specific application domains. Moreover in [16] the impact of a specific lexical resource has been compared with the one achievable with lexical information gathered through the distributional analysis of a large scale corpus. Experimental findings show that a distributional approach provides better results. This is very interesting as distributional approaches are unsupervised and largely applicable directly from the application domain texts. However a proper investigation of the impact of different possible geometrical representations is still needed. By employing different notions of *context*, we can assume two words similar when they appear *in the same documents* [18, 19] or *in the same sentences* (modeled as word co-occurrences in short windows [2]) or even *in the same syntactic structures* [8].

In this work, we investigate how the choice of the above different semantic representations impacts on the generalization capability of the SPTK. First, a direct evaluation of distributional models is carried out in the Semantic Text Similarity (STS) task [20], where the measure of semantic similarity between sentence pairs is tackled.

An additional indirect evaluation is discussed in a fine-grained semantic task, i.e. the Question Classification (QC) task. In the rest of the paper, Sect. 2 discusses the impact of different semantic representations. In Sect. 3 different Convolution Kernels among linguistic structures will be discussed. Section 4 evaluates the impact of different semantic representations in STS and QC tasks. Section 5 derives the conclusions.

2 Distributional Models of Lexical Semantics

Distributional approaches represent lexical semantics through the analysis of observations in large-scale corpora. The fundamental intuition is that the meaning of a word can be described by the set of textual contexts in which it appears. It is commonly known as *Distributional Hypothesis* [1] and can be synthesized from the following statement in [21]:

> Words with similar meanings will occur with similar neighbors if enough text material is available.

The idea is to acquire an artificial representation of a target word w, considering all other words co-occurring with w, such that two words sharing the same co-occurrences will be represented in a similar manner. A lexical similarity function can be thus defined in terms of similarity between these representations. Notice that a good approximation of the words distributional information can be achieved if a sufficient amount of observations is gathered. Several large scale corpora can be exploited in English, e.g. the British National Corpus (BNC) [22] made of 100 million words, the GigaWord [23], made of 1.75 billion words, or the ukWaC corpus [24], made of 2 billions word. Other corpora are available also for other languages, e.g. itWaC a 2 billions word for Italian.

Within this study, a distributional representation of words is acquired through a high-dimensional space known as *Word Space*, where the distance among instances (i.e. words) reflects the lexical similarity, as described in [25]:

> Vector similarity is the only information present in Word Space: semantically related words are close, unrelated words are distant.

Words are point in this space and if two words have similar contexts, they will have similar representations and they will be close in the space. From a linguistic perspective, they are likely to be related by some type of generic semantic relation, either paradigmatic (e.g. synonymy, hyperonymy, antonymy) or syntagmatic (e.g. meronymy, conceptual and phrasal association), as observed in [2].

Semantic spaces have been widely used for representing the meaning of words or other lexical entities, as discussed in [3, 8, 26], with successful applications in lexical disambiguation, as in [4], harvesting thesauri, as in [27] and Name Entity Classification, as in [6].

From a computational perspective, a matrix M is defined, whose rows describe words as vectors $\boldsymbol{w_i}$, columns describe the corpus contexts $\boldsymbol{c_j}$ and each entry w_{ij} is a measure associating words and contexts. Given two words w_1 and w_2, the term similarity function can be estimated according to the Euclidean Distance or the *Cosine Similarity* between the corresponding projections $\boldsymbol{w_1}$, $\boldsymbol{w_2}$, i.e.

$$\cos(w_1, w_2) = \frac{\boldsymbol{w_1} \cdot \boldsymbol{w_2}}{\|\boldsymbol{w_1}\| \|\boldsymbol{w_2}\|} \tag{1}$$

that measures the angle between such vectors.

One open issue is that a definition of c_j and the association measure's estimation has not yet been addressed. This problem is not trivial as different semantic aspects of the involved words are considered by changing the representation space. For example, by employing different notions of context, we can assume two words similar when they appear *in the same documents* [18, 19] or *in the same sentences* (modeled as word co-occurrences in short windows [2]) or even *in the same syntactic structures* [8]. Obviously, different context types define geometric spaces with different semantic properties and different generalization grains in the resulting similarity estimation.

Moreover, different NLP tasks require different types of lexical generalization. A wider context will provide a shallower generalization while a smaller one will capture more specific lexical aspects of words, as well as their syntactic behavior.

2.1 Different Word Spaces for Different Lexical Relations

In a typical Information Retrieval task, i.e. document classification task, where the aim is to map each document in a class reflecting the text topic (e.g. sport, economy or science), a topic-oriented form of similarity (i.e. topical similarity) is required. Such a model can be employed to relate words like "bank", "acquisition", "purchase" "money" or "sell", as they address one single "economic" topic. On the contrary, paradigmatic relations could be more appropriate for other tasks. In the FrameNet based Semantic Role Labeling task, if one needs to infer that "knife" refers to the `Instrument` role in a sentence like "Mary killed John with a knife", a more specific notion of similarity is needed to relate this sentence with some prior knowledge, e.g. that "rifle" or "knife" are examples of INSTRUMENT within the KILLING situation. In the following, three different kinds of context are investigated.

Topical Space. A document-based geometric space represents words by focusing on coarse grain textual elements, capturing contextual information by expressing the distribution of words across documents [18]. Two words will have a similar geometric representation if they tend to appear in the same documents of a corpus. In Information Retrieval this notion is usually employed to represent texts via linear combinations of (usually orthonormal) vectors corresponding to their component words. This space can be computationally represented as a so-called word-by-document matrix having as many rows as the number of (unique) target words to represent, and having as many columns as the number of different documents in the underlying corpus. Individual scores, associating words and documents, are computed according the term frequency-inverse document frequency (tf-idf) schema [18]. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others. As a results, two words tending to occur in the same documents will have a similar set of active components (i.e. value in the same columns). In such way words like *bank* or *acquire* have the same representation because they tend to appear in documents concerning the same economical topics, thus sharing a topical relation.

Word-based Space. This particular space aims at providing a distributional lexical model while capturing paradigmatic relations between target words *tw*s. Paradigmatic relations concern substitution, and relate entities that do not co-occur in the text. It is a relation *in absentia* and holds between linguistic entities that occur in the same context but not at the same time, like the words *knife* and *rifle* in the sentence "to kill with a [knife|rifle]". Paradigmatic relations are substitutional relations, which means that linguistic entities have a paradigmatic relation when the choice of one excludes the choice of another. A paradigm is thus a set of such substitutable entities.

In a Word-based space, vectors represent *tw*s, while dimensions are words appearing in a *n*-windows around the *tw*s [2]. To better understand, let us consider the adjectives *beautiful*, *attractive* and *pretty*. They are synonyms, i.e. words that can be mutually exchanged in texts, in most cases without altering the corresponding meaning, e.g. in phrases like "the beautiful girl", "the attractive girl" or "the pretty girl". Just considering these simple examples, we can notice that these words co-occur with the word *girl*. If synonyms can be exchanged in the language in use, in a large-scale document collection they will tend to co-occur with the same words. If vector dimensions correspond to words in the corpus, in a Word-based space *tw*s co-occurring with the same set of words are similarly represented, having initialized almost the same set of geometrical components. This is not valid only for synonyms, as words involved in a paradigmatic relation have the same properties. If two words like *knife* of *rifle* can be exchanged in texts, they share a consistent subset of co-occurring words.

Then, in this words-by-words matrix each item is a co-occurrence count between a *tw* (a row) and other words in the corpus, within a given window of word tokens. The window width *n* is a parameter allowing the space to capture different lexical properties: larger values for *n* tend to introduce more words, i.e. possibly noisy information, whereas lower values lead to sparse representations more oriented to paradigmatic properties. Moreover, in order to capture a first form of syntactic information, words co-occurring on the left are treated separately from words co-occurring on the right. It allows, for example, to provide a better representation for transitive or intransitive verbs. In a sentence like "the beautiful girl entered the bar", we say that *beautiful* co-occurs with *the* in a left widow of size one, with *girl* in a right window of size one, with *entered* in a right window of size two, with *the* in a right windows of size three and *bar* in a right window of size four. To provide a robust weighting schema and penalize common words, whose high frequency could imply an unbalanced representation, Pointwise Mutual Information (PMI) [3, 28] scores are here adopted. In order to make words representation more sensitive to their syntactic behavior, the Word-based Space model can be easily extended by differently considering contextual words, depending on the side of their co-occurrences.

Syntax-based Space. Finally, the Syntax-based space aims at capturing paradigmatic relations as well as the Word-based space, but imposing more strict syntactic constraints over the context selection. This distributional space is enriched by features directly expressing syntactic information, as discussed in [8]. A syntactic analysis of the entire corpus is required and the dependency formalism is here employed. An example of dependency parse tree associated to the sentence "The beautiful girl entered the bar" is shown in Fig. 1.

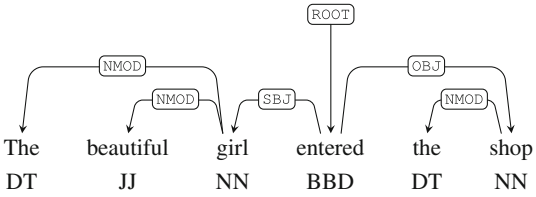

Fig. 1 Example of a dependency parse tree

The words-by-words matrix here records the number of times a *tw* (i.e. the rows) co-occurs with another word w in a specific syntactic relation $\mathtt{r}$. Columns are thus corresponding to word-relation pairs, so that each space dimension reflects the pair $\langle \mathtt{r}, w \rangle$. For example, given the verb *entered* in Fig. 1, it records the number of times is directly connected to other words, such as $\langle \mathtt{SUBJ}, girl \rangle$ and $\langle \mathtt{OBJ}, bar \rangle$. It allows to consider two co-occurring words irrespectively of whether they are physically adjacent or not. At the same time syntactic relations embed information derived from complex linguistic structures, such as argument-structure (e.g., subject-verb, verb-object, verb-indirect object) or modification (e.g., adjective-noun, noun-noun), as discussed in [8]. Vector components here provide a more precise representation as a lot of (possibly noisy) material is filtered out, although resulting in a very sparse representation. The individual score is computed according the Pointwise Mutual Information (PMI) [3, 28], as for the Word-based space.

2.2 Embedding Lexical Semantics in Lower Dimensional Spaces

The quality of a Word Space is tied to the amount of information analyzed: the more contextual information is provided, the more accurate will be the resulting lexical representation. However, some problems of scalability arise when the number of the space dimension increases. From a computational perspective, a space with thousands of dimensions makes the similarity estimation between vectors expensive. Consequently, even a simple operation, e.g. the search of the most similar words to a target word, can be prohibitive. Moreover, from a geometric perspective, the notion of similarity between vectors is sparsely distributed in a high-dimensional space. This is known as the *curse of dimensionality*, as discussed in [29]: in this scenario, the higher the number of dimensions is, the lower is the variance of distances among data, reducing the expressiveness of this information for further inferences.

Fortunately, employing geometric representation for words enables the adoption of *dimensionality reduction techniques* to reduce the complexity of the high-dimensional space. Such techniques allow to exploit data (i.e. words and contexts) distribution and topology in order to acquire a more compact representation and more meaningful data-driven metrics. The main differences between techniques for dimensionality reduction are in the distinction between *linear* and *nonlinear*

methodologies. Linear techniques assume that the data lie on a linear (or near linear) subspace whose dimensions are smaller than the original space. Nonlinear techniques instead assume that data lie on an embedded non-linear *manifold* within the higher-dimensional space [30].

Latent Semantic Analysis [19] is an example of linear dimensionality reduction technique and uses the Singular Value Decomposition (SVD) [31] to find the best subspace approximation of the original word space, in the sense of minimizing the global reconstruction error projecting data along the directions of maximal variance. The original word-by-context matrix M is decomposed through SVD into the product of three new matrices: U, S, and V so that S is diagonal and $M = USV^T$. M is approximated by $M_k = U_k S_k V_k^T$ in which only the first k columns of U and V are used, and only the first k greatest singular values are considered. This approximation supplies a way to project a generic term w_i into the k-dimensional space using $W = U_k S_k^{1/2}$, where each row $\boldsymbol{w}_i^k$ corresponds to the representation vectors $\boldsymbol{w_i}$. The original statistical information about M is captured by the new k-dimensional space which preserves the global structure while removing low-variant dimensions, i.e. distribution noise. The lexical similarity can still be computed in such reduced space with the cosine similarity expressed in Eq. 1 in a space with a reduced number of dimensions (e.g. $k = 100$) where the notion of distance is more significant with respect to the original space. These newly derived features may be considered latent concepts, each one representing an emerging meaning component as a linear combination of many different original contexts.

It is worth noticing here that the application of SVD to different spaces results in very different latent topics. The emerging of special directions in the space as caused by different linguistic contexts (e.g. from documents to short windows around words) has thus significantly different linguistic implications. When large contexts are used, the resulting latent topics act as primitive concepts to characterize document topics, i.e. aspects of the domain knowledge related to the corpus. When short contexts are adopted in M, latent topics characterize primitive concepts needed to distinguish[1] short phrases: they thus tend to capture paradigmatic word classes, for which syntactic substitutability holds. In order to determine lexical information provided by the proposed distributional models, an empirical analysis of the latent semantic topics obtained by SVD over the different source spaces, i.e. topical, word-based and syntactic-based space, has been carried out, in order to find the possible different generalizations obtained in these cases. Different distributional models are acquired from the ukWaC [24] corpus, a large scale Web document collection made by 2 billion tokens. All *tw*s occurring more than 200 times (i.e. more than 50,000 words) are represented and different approaches discussed above are applied as follows to define the word-by-context matrix M:

[1] Note that SVD emphasizes directions with maximal covariance for M, i.e. term clusters for which it is maximal the difference between contexts, i.e. short syntagmatic patterns.

- **Topical Space**: the entire corpus has been split so that each column of M represents a sentence. The number of different sentences is about 1,500,000 and each matrix item contains the *tf-idf* score of a target word (*tw*) with respect to each corresponding sentence. It means that contexts, i.e. the matrix columns, are sentences in the ukWaC corpus and two words are related if they tend to co-occur in the same sentences.
- **Word-based Space**: a co-occurrence word-based space provides a more specific notion of similarity and contexts are not sentences anymore, but instead other words in the corpus. It means that two words are related if they co-occur with other words in the ukWaC corpus in a window of size $n = 3$. This particular context dimension is selected to have a more precise representation and better capturing paradigmatic relations between words. Individual co-occurrence scores are weighted according to the Pointwise Mutual Information (PMI), as estimated in [3].
- **Syntax-based Space**: contexts are made of syntactically-typed co-occurrences in dependency graphs built from the entire set of the ukWaC parsed sentences through the LTH parser [32]. The most frequent 150,000 basic features, represented as the ⟨`synt_rel`, *lemma::pos*⟩ pair, are employed as contextual features corresponding to PMI scores.

The SVD reduction is finally applied to each matrix M with a dimensionality cut of 250. This empirical evaluation consists in the projection of a noun and a verb, i.e. *ruler.n* and *defeat.v*, into the reduced space and the selection of the most five similar words according to the cosine similarity measure, expressed in Eq. 1. By projecting the noun *ruler.n* in the topical space, the five most similar words are *persia.n*, *persian.n*, *rebels.n*, *dominium.n*, *medes.n*, in the word-based space are *conqueror.n, emperor.n, dominium.n, dynasty.n* and *tyrant.n*, while in the syntax-based space are *emperor.n*, *monarch.n*, *overlord.n* and *dictator.n*. For the verb *defeat.v* the most similar words are *fight.v, lieutenant-colonel.n, knight.n, whip.n* and *wavell.n* according to the topical space, *victory.n, defeat.n, overthrow.v, victorious.j* and *fight.v* according to the word-based space and *beat.v*, *fight.v*, *conquer.v*, *oust.v* and *overthrow.v* according to the syntax-based space. The example seems to show that paradigmatic generalizations are captured in the word-based space, whereas *ruler.n* and *defeat.v* are correctly generalized in synonyms (such as *emperor.n/dominium.n* and *overthrow.v/fight.v*, respectively). The document space instead seems to suggest topical similarity (such as *persian.n* versus *ruler.n* or *knight.n,whip.n* versus *defeat.v*) that tends to relate words at a broader level. Finally, the syntactic space seems to capture paradigmatic relations as well as the word-based space, moreover imposing syntactic constraints over the word selection: this phenomenon is particularly evident in the analysis of *defeat.v* where only transitive verbs have been selected. It is not trivial to provide a judgment on the best space for every language learning task, as different semantic relations between lexemes, i.e. topical or paradigmatic relations, may contribute differently depending on the target problem.

3 Kernel-Based Learning and Distributional Information

In kernel-based machines, both learning and classification algorithms only depend on the inner product between instances. If an example is not represented by a vector, the product can be implicitly computed by kernel functions by exploiting the following dual formulation: $\sum_{i=1,\ldots,l} y_i \alpha_i \phi(o_i)\phi(o) + b = 0$, where o_i and o are two objects, ϕ is a mapping from the objects to feature vectors $\boldsymbol{x_i}$ and $\phi(o_i)\phi(o) = K(o_i, o)$ is a kernel function, as discussed in [33]. The function K can be applied to any algorithm which solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced by K. As a side effect, the so-called *Kernel Trick*, those candidate linear algorithms are transformed into non-linear algorithms. Those non-linear algorithms are equivalent to their linear counterparts operating in the range space of a feature space determined by $\phi(\cdot)$. In other words, a kernel function, allows us to express the similarity between two objects, that are explanatory of the target problem, without defining their explicit representation.

In this section different kernels among tree structures, i.e. Tree Kernels, will be discussed. Then geometrical models of lexical semantics will be presented according to a kernel perspective. Finally the formulation of a kernel, able to combine syntactic and lexical information, i.e. the Smoothing Partial Tree Kernel will be discussed.

3.1 Convolution Tree Kernels

Convolution Tree Kernels (TK) compute the number of substructures that are common between two trees T_1 and T_2, without explicitly considering the whole fragment space. For this purpose, let the set $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ be a tree fragment space and $\chi_i(n)$ be an indicator function, equal to 1 if the target f_i is rooted at node n and equal to 0 otherwise. A tree-kernel function over T_1 and T_2 is $TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$, N_{T_1} and N_{T_2} are the sets of the T_1's and T_2's nodes, respectively and

$$\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \chi_i(n_1)\chi_i(n_2)$$

The latter is equal to the number of common fragments rooted in the n_1 and n_2 nodes. The Δ function determines the richness of the kernel space.

A largely known kernel, i.e. Syntactic Tree Kernel (STK), has been introduced in [34] to define a similarity between two sentences by exploiting their syntactic structures, i.e. the parse trees. It is sufficient to compute $\Delta_{STK}(n_1, n_2)$ as follows (recalling that since it is a syntactic tree kernels, each node can be associated with a production rule): (i) if the productions at n_1 and n_2 are different then $\Delta_{STK}(n_1, n_2) = 0$; (ii) if the productions at n_1 and n_2 are the same, and n_1 and n_2 have only leaf children then

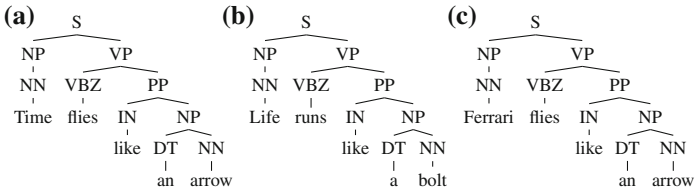

Fig. 2 Examples of syntactic parse trees

$\Delta_{STK}(n_1, n_2) = \lambda$; and (iii) if the productions at n_1 and n_2 are the same, and n_1 and n_2 are not pre-terminals then $\Delta_{STK}(n_1, n_2) = \lambda \prod_{j=1}^{l(n_1)}(1 + \Delta_{STK}(c_{n_1}^j, c_{n_2}^j))$, where $l(n_1)$ is the number of children of n_1 and c_n^j is the j-th child of the node n.

It is a very powerful method as it counts the common subtree structures shared by the sentences in a implicit space where each component corresponds to one possible tree fragment. Figure 2 shows the parse trees of the sentences *Time flies like an arrow*, *Life runs like a bolt* and *Ferrari flies like an arrow*, respectively. Common subtrees that would contribute to a kernel would be `(S(NP)(VP))` or `(NP(DT)(NN))`. The main advantage is that it is not necessary to define explicitly all the possible tree configurations, as only components useful to estimate the similarity will be taken into account. In this implicit space the resulting vector can be seen as the composition of all the atomic information (i.e. the tree fragments) needed to reflect the syntactic structure of the sentence, as well the lexical information of the tree leaves (i.e. the words). STK are rigid measures of semantic similarity as for their strict requirements on the matching of syntactic substructures. The tree kernel discussed in [34] only triggers matches that fully satisfy derivation rules in the underlying grammars: this implies that only identical words appearing in the corresponding syntactic position are matched.

Partial Tree kernels (PTK, [35]) are an attempt to relax these grammatical constraints, but they only act at the syntagmatic level. If a partial match between two syntactic structures is applied, the corresponding skipped material is fully neglected. It does not provide any contribution to the kernel, i.e. no lexical contribution can be observed. The computation of PTK is carried out by the following Δ_{PTK} function: if the labels of n_1 and n_2 are different then

$$\Delta_{PTK}(n_1, n_2) = 0;$$

else

$$\Delta_{PTK}(n_1, n_2) = \mu\left(\lambda^2 + \sum_{\boldsymbol{I}_1, \boldsymbol{I}_2, l(\boldsymbol{I}_1)=l(\boldsymbol{I}_2)} \lambda^{d(\boldsymbol{I}_1)+d(\boldsymbol{I}_2)} \prod_{j=1}^{l(\boldsymbol{I}_1)} \Delta_{PTK}(c_{n_1}(\boldsymbol{I}_{1j}), c_{n_2}(\boldsymbol{I}_{2j}))\right) \quad (2)$$

where $d(\boldsymbol{I}_1) = \boldsymbol{I}_{1l(\boldsymbol{I}_1)} - \boldsymbol{I}_{11}$ and $d(\boldsymbol{I}_2) = \boldsymbol{I}_{2l(\boldsymbol{I}_2)} - \boldsymbol{I}_{21}$. This way, we penalize both larger trees and child subsequences with gaps.

3.2 Smoothing Partial Tree Kernels

Combining lexical and structural kernels provides clear advantages on all-vs-all words similarity, which tends to semantically diverge. Indeed syntax provides the necessary restrictions to compute an effective semantic similarity. Following this idea, Bloedhorn and Moschitti [15] modified step (i) of Δ_{STK} computation as follows: (i) if n_1 and n_2 are pre-terminal nodes with the same number of children, $\Delta_{STK}(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} \sigma(lex(n_1), lex(n_2))$, where lex returns the node label. This allows to match fragments having same structure but different leaves by assigning a score proportional to the product of the lexical similarities of each leaf pair. Although it is an interesting kernel, the fact that lexicals must belong to exactly the same structures and on the leaf nodes limits its applications. As described in [16], a smoothed tree kernel, that can be applied to any tree, exploits a lexical semantic kernel, while respecting the syntax enforced by the tree. When a partial tree kernel is employed, i.e. Eq. 2, its smoothed counterpart is defined as follows: if n_1 and n_2 are leaves then $\Delta_\sigma(n_1, n_2) = \mu\lambda\sigma(n_1, n_2)$; else

$$\Delta_\sigma(n_1, n_2) = \mu\sigma(n_1, n_2) \times \left(\lambda^2 + \sum_{\boldsymbol{I}_1, \boldsymbol{I}_2, l(\boldsymbol{I}_1)=l(\boldsymbol{I}_2)} \lambda^{d(\boldsymbol{I}_1)+d(\boldsymbol{I}_2)} \prod_{j=1}^{l(\boldsymbol{I}_1)} \Delta_\sigma(c_{n_1}(\boldsymbol{I}_{1j}), c_{n_2}(\boldsymbol{I}_{2j})) \right) \tag{3}$$

where σ is any lexical similarity between nodes (i.e. Eq. 1) and the other variables are the same as in PTK, Eq. 2. We call this kernel a *smoothed partial tree kernel*, i.e. *SPTK*. In this formulation, for every tree pair the σ function estimates the similarity among the nodes, so if labels are the same (i.e. $\sigma = 1$) the contribution is equal to $\mu\lambda^2$, as in the PTK; otherwise the contribution of the nodes and the subtrees is weighted accordingly to the information provided by the word space,[2] whose quality is crucial. If σ tends to confuse words not semantically related or apply too much smoothing, the overall learning algorithm will not be able to well characterize useful examples. A too strict function will otherwise produce the same results of a pure PTK.

4 Experimental Evaluation

The aim of the experiments is to measure how different grammatical representations, i.e. dependency structures, and different lexical semantic representations impact on the effectiveness of the *SPTK* kernel. Accordingly, we carried out extensive

[2] When n_1 and n_2 are not lexical nodes σ will be 0 when $n_1 \neq n_2$.

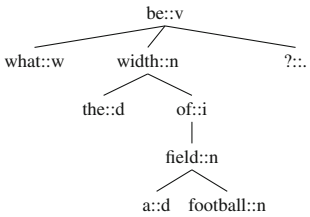

Fig. 3 Lexical Only Centered Tree (LOCT)

experiments on Semantic Text Similarity (STS) and Question Classification (QC), as a specific, yet complex, semantic inference.

4.1 General Experimental Setup

According to findings discussed in [16], the SPTK achieves best results when applied to structures obtained from dependency parse trees. Sentences are parsed with the LTH dependency parser described in [32].

Figure 3 shows the Lexical Only Centered Tree (LOCT) which is directly derived by the parse tree. It only accounts on the lexicals, where untyped binary relations are used for recursive structures. The grammatical generalization provided by the syntactic edge labels is thus neglected. In the empirical perspective pursued here, this structure is interesting as the lexical generalization is applied to all tree nodes corresponding to content words. We apply lemmatization to the lexicals to limit sparseness and, at the same time, we also adopt a set of 10 simplified PoS-tags, e.g. noun (n::), verb (v::), adjective (::a). This allows to measure similarity only between lexicals in the same grammatical category. In contrast, the LCT shown in Fig. 4 represents the dependency structures where both grammatical and all PoS-Tags are retained as rightmost children.

The corpus employed to develop the word spaces, i.e. ukWak [24], is a large scale document collection made by 2 billion tokens. To reduce data sparseness all target words tws that occur in the ukWak more than 200 times have been selected, i.e. more

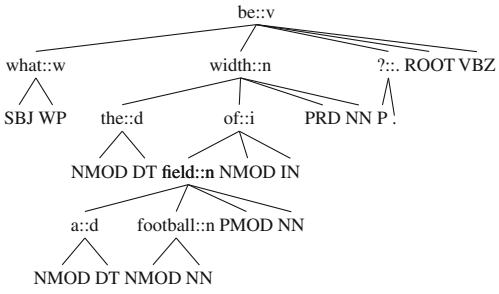

Fig. 4 Lexical Centered Tree (LCT)

that 50,000 words. Each tw corresponds to a matrix row and it is labeled with the pair $\langle lemma, ::POS \rangle$. Then different approaches are applied to build the word-by-context matrix M, as described in Sect. 2.1:

Topical Space: the entire corpus has been split so that each column of M represents a sentence. The number of different sentences is about 1,500,000 and each matrix item contains the *tf-idf* score of tw with one corresponding sentence.

Word-based Space win. $\boldsymbol{n}$: for this co-occurrence word space, left contexts are treated differently from the right ones. Each column of M represents a word in the corpus and each item measures the number of times this word co-occurs with tw in a window of size $\pm n$. The most frequent 20,000 items are selected, so that M models 40 k contexts (i.e. right and left contexts). Two window sizes are employed: a size $n = 3$ to have a more precise representation and better capturing syntactic properties of words and $n = 6$ to provide a more smoothed generalization.

Syntax-based Space: contexts here are made of syntactically-typed co-occurrences within dependency graphs built from the entire set of ukWak sentences. This is a very sparse space and the most frequent 150,000 basic features, i.e. ⟨`synt_rel`, *lemma::pos*⟩, are employed as contextual features corresponding to PMI scores. The SVD reduction is then applied to M, with three different dimensionality cuts of 30, 100 and 250, where a lower dimensionality provides a larger compression but a less precise generalization. We experiment with multi-classification, which we model through *one-vs-all* scheme by selecting the category associated with the maximum SVM margin. In all the experiments the kernel estimation is normalized. To have a normalized similarity score between 0 and 1, given two trees T_1 and T_2 a normalization in the kernel space is applied as $\frac{TK(T_1,T_2)}{\sqrt{TK(T_1,T_1)\times TK(T_2,T_2)}}$.

4.2 Semantic Text Similarity: Results

In this first experiment the contribution of different distributional models of lexical semantics is evaluated within the measure of semantic relatedness between entire sentences. In particular we targeted the Semantic Textual Similarity (STS) task proposed in [20]. Similarity scores between sentence pairs are provided by annotators: scores range between 0 (uncorrelated pairs) and 5 (identical pairs). Competing systems are asked to provide scores (not necessarily in the same range) whereas performances are measured through the *Pearson Correlation* with respect to human judgments. As text similarity strictly depends on the similarity at lexical level as well as on the equivalence of more complex syntagmatic structures, the STS is ideal for evaluating the impact of a semantic similarity measure in a realistic setting. In the STS challenge, four datasets made of sentences derived from different corpora and modeling different aspects of similarity have been provided as test datasets: the *headlines* dataset include headlines mined from several news sources by European Media Monitor using the RSS feed; in the *OnWN* dataset, sentences are sense definitions from WordNet and OntoNotes [36]; in the *FNWN* dataset, sentences are sense definitions

from WordNet and FrameNet; finally, the *SMT* dataset comes from DARPA GALE HTER and HyTER, where one sentence is a MT output and the other is a reference translation where a reference is generated based on human post editing (provided by LDC) or an original human reference (provided by LDC) or a human generated reference based on FSM.

A first similarity function is obtained without accounting for the syntactic composition of the lexical information involved in the sentences. Basic lexical information is obtained by different distributional models. Every word appearing in a sentence is then projected in such space and a sentence can be represented by applying an additive linear combination in the Latent Semantic space, as described in Sect. 2.2. The similarity function between two sentences is then the cosine similarity between their corresponding vectors, namely the Latent Semantic Kernel (LSK), in line with [37].

Then, the SPTK is applied to the LCT representation derived from the dependency parse tree. Moreover, we also measured the contribution of SPTK with respect to the traditional Partial Tree Kernel (PTK) [35]. In fact, without considering any specific similarity function between lexical nodes the SPTK can be considered as a PTK that captures a more strict syntactical similarity between texts.

Table 1 shows result in term of Pearson Correlation between the human judgment and the score provided by different kernels. We did not report any comparison with the best results of the SemEval STS competition as those approaches are mostly supervised. On the contrary the presented approach for the STS estimation is fully unsupervised. In the FNWN and OnWN datasets, best results are achieved when using the Word-based Space, so capturing paradigmatic relations among words, in combination with the LSK operator, so neglecting the sentence syntactic structure. It is reasonable as both datasets provide definitions, and syntax is not informative as all sentences have similar declarative forms. This is confirmed by the poor results achieved by the PTK, that is not able to separate sentence similarities. It is slightly different for the headlines, where news are targeted and a topical similarity is more competitive, so capturing the main theme of the described event. Here syntax is more important as provided by higher results achieved by the PTK and, even more,

Table 1 Results provided by the different distributional models within the STS task

	FNWN			OnWN		
	LSK	PTK	SPTK	LSK	PTK	SPTK
Topical	0.337		0.303	0.597		0.506
Word-based (win. 3)	**0.448**	0.047	0.371	**0.646**	0.275	0.527
Syntax-based	0.431		0.412	0.607		0.540
	Headlines			SMT		
	LSK	PTK	SPTK	LSK	PTK	SPTK
Topical	0.595		0.631	0.235		0.331
Word-based (win. 3)	0.596	0.472	0.637	0.294	0.276	0.331
Syntax-based	0.574		0.635	0.285		0.348

by the SPTK. While the contribution of specific spaces is not relevant in the headlines dataset, results within the SMT dataset benefit by the strict information provided by the Syntax-based: the main reason is that many similar sentence pairs, such as "things about others, say them carefully" and "issues about others, discuss them cautiously" have similar syntactic structures, but different lexemes.

4.3 Question Classification: Results

For these experiments, we used the UIUC QC dataset [38], made by a training set of 5,452 questions and a test set of 500 questions.[3] Question classes are organized in two levels: 6 coarse-grained classes (like `ENTITY` or `HUMAN`) and 50 fine-grained sub-classes (e.g. `Plant`, `Food` as subclasses of `ENTITY`). While the former is more sensitive to syntax, the latter is highly dependent on lexical information. Giving the particularly limited number of training examples available for the individual fine-grained classes, the lexical generalization acquired from the external corpus through distributional analysis is crucial. We employed the SVM learning algorithm, by extending the SVM-LightTK software[4] [35] with the *SPTK* defined by Eq. 3.

The quality of such classification is measured with accuracy, i.e. the percentage of test examples that are correctly classified. The parametrization of each classifier is carried on a held-out set (30 % of the training) and concerns with the setting of the trade-off parameter (option `-c`). This fix split between train and test is useful to have a more meaningful comparison between the different employed spaces. Moreover, it is the same experimental setup provided in [38]. In contrast, the cost-factor parameter of the SVM-LightTK (option `-j`) is set as the ratio between the number of negative and positive examples, for attempting to get a balanced contribution of training examples.

In these experiments, a model that does not account on the syntactic structure of the sentence (i.e. a Bag of Word model) is employed as baseline. When only lemmatized words are considered and no SVD reduction is applied, an accuracy of 89.4 and 83.8 % for the coarse-grained and fine-grained setting is estimated respectively. The outcome of the several kernels applied to several structures for the coarse and fine-grained QC is reported in Tables 2 and 3 respectively. The first column shows different experimented spaces employed with the SPTK. The second column refers to different dimensionality reduction via SVD employed. The last two columns report the accuracy scores obtained by applying the *SPTK* kernel to the LOCT and LCT structures of Sect. 4.1. The first line contains accuracy where no generalization is applied, i.e. kernel formulation is comparable with the PTK described in [35].

In the coarse grain setting, best results are obtained by the LCT structure where the improvement of 4 % in accuracy (from 90.8 to 94.8 %) confirms that the lexical generalization is very useful even for tasks like the coarse grained QC, for which the syntactic structure of the question is the most discriminative feature. This is

[3] http://cogcomp.cs.illinois.edu/Data/QA/QC/.

[4] http://disi.unitn.it/moschitti/Tree-Kernel.htm.

Table 2 Accuracy of structural kernels for coarse grained QC

Space	Dimens.	LOCT (%)	LCT (%)
–	–	89.2	90.8
Topical	30	71.8	86.8
	100	85.8	91.4
	250	88.6	92.0
Word-based (win. 3)	30	86.6	93.4
	100	90.4	94.4
	250	93.6	**94.8**
Word-based (win. 6)	30	89.4	92.2
	100	92.8	93.6
	250	93.0	93.8
Syntax-based	30	86.4	91.8
	100	91.6	94.0
	250	94.2	93.8

Table 3 Accuracy of structural kernels for fine-grained QC

Space	Dimens.	LOCT (%)	LCT (%)
–	–	85.4	85.4
Word-based (win. 3)	30	79.2	82.4
	100	85.8	85.2
	250	87.2	86.8
Word-based (win. 6)	30	80.0	80.6
	100	85.4	85.0
	250	**87.4**	86.6
Syntax-based	30	71.2	78.4
	100	84.4	84.2
	250	87.2	86.4
Topical	30	59.2	76.6
	100	81.2	81.8
	250	84.0	84.6

confirmed by results achieved by the LCT that, although using explicit syntactic labels, outperform the LOCT. It is worth to notice that best results are obtained by the co-occurrence word space with a window size of three, thus confirming the need of a specific generalization for lexicals. As already noticed in Sect. 2.1, different word spaces seem to capture different linguistic generalizations. The co-occurrence word space outperforms the Syntactic Word Space (94.8 % respect to 93.8 %). It suggests that, while the latter space is more precise, its overall accuracy can be reduced by parsing errors as well as by data sparseness, as every component in the space corresponds to a word typed by a syntactic relation. This finding is also confirmed

in the fine-grained setting, where the impact of a co-occurrence word space is more beneficial. The fine-grained setting represents a task in which the lexical information is much more effective. The LOCT representation here achieves the best results (i.e. 87.4 %) although the differences among word spaces are negligible. Not every space overcomes the baseline. The topical space, in both settings, is quite unstable, especially for LOCT, where no explicit syntax is encoded in the tree. This is in line with the assumption that applying SVD over document-based spaces results in domain (or topical) similarity that is a rather different notion than paradigmatic similarity. As the *SPTK* kernel requires a semantic smoothing harmonic with the syntax, paradigmatic relations are preferable, as they better comply to substitutability in interpretation. These latter relations seem to be better captured by co-occurrence word spaces with smaller windows, as the difference in performance between $n = 3$ and $n = 6$ suggests. As an example, a question whose classification is mistaken by the bag-of-word approach, as well as by the PTK (with no lexical smoothing) is `Q`: *What French ruler was defeated at the battle of Waterloo?* Also the classification with a *SPTK* built over a topical space wrongly associates `Q` with `ENTITY` rather than with the correct coarse category of `HUMAN`. It is clearly an example of a question whose lexical information needs to be generalized to induce that a *ruler* is a man as it can be *defeated*.

An error analysis shows the contribution of *SPTK* for sentences like "What peninsula is Spain part of?" or "What French ruler was defeated at the battle of Waterloo?". Without knowing that *peninsula* indicates a geographic location (or "ruler" a person) the most probable category could be ENTITY. In contrast, SPTK can provide the correct answer by measuring, for example, the structural similarity of the first question with the training question: *What island group is Guadalcanal a part of?* along with the lexical similarity between *peninsula* and *island* and between *Spain* and *Guadalcanal*.

5 Conclusion

In this work an extensive study of the role of vector space approaches to lexical meaning in tree kernel based natural language learning has been carried out over a the Semantic Text Similarity (STS) and Question Classification (QC) tasks. The lexical generalization provided by the word space approaches is always beneficial with significant performance improvements in coarse as well as fine-grained QC tasks. However, not all the vector spaces are equivalently useful, when they are employed as generalization functions for tree kernels. While document oriented representations are not always well suited to support the required lexical generalizations, word spaces with smaller co-occurrence windows seem to capture paradigmatic relations that are quite useful. The improvements achieved in this paper are remarkable. They are in fact providing a novel state-of-the-art on a well known task (i.e. QC) also successfully tackled by previous, more complex, models. This is inline with the results achieved in [16]. Future work will investigate if the beneficial role of proper

lexical generalizations in SPTKs is also observable in other tasks, e.g. Semantic Role Labeling. The general outcome of this work suggests that vector representations are not all equally expressive of the variety of semantic relations they tend to capture, and their employment in semantic NLP tasks must be carefully designed.

References

1. Harris, Z.: Distributional structure. In: Katz, J.J., Fodor, J.A. (eds.) The Philosophy of Linguistics. Oxford University Press, Oxford (1964)
2. Sahlgren, M.: The word-space model. PhD thesis, Stockholm University (2006)
3. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010)
4. Schutze, H.: Automatic word sense discrimination. J. Comput. Linguist. **24**, 97–123 (1998)
5. Lin, D.: Automatic retrieval and clustering of similar word. In: Proceedings of COLING-ACL, Montreal, Canada (1998)
6. Giuliano, C.: Fine-grained classification of named entities exploiting latent semantic kernels. In: Proceedings of CoNLL 2009, CoNLL'09, Stroudsburg, PA, USA, pp. 201–209 (2009)
7. Croce, D., Giannone, C., Annesi, P., Basili, R.: Towards open-domain semantic role labeling. In: ACL, pp. 237–246 (2010)
8. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. Comput. Linguist. **33**(2) (2007)
9. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cogn. Sci. **34**, 1388–1429 (2010)
10. Baroni, M., Lenci, A.: One distributional memory, many semantic spaces. In: Proceedings of the GEMS 2009 Workshop, GEMS'09, Stroudsburg, PA, USA, pp. 1–8 (2009)
11. Clark, S., Pulman, S.: Combining symbolic and distributional models of meaning. In: Proceedings of the AAAI Spring Symposium on Quantum Interaction, pp. 52–55 (2007)
12. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. In: Proceedings of EMNLP 2011, Edinburgh, Scotland, UK
13. Haussler, D.: Convolution kernels on discrete structures. University of Santa Cruz, Technical report (1999)
14. Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In: Proceedings of ACL'02 (2002)
15. Bloehdorn, S., Moschitti, A.: Combined syntactic and semantic kernels for text classification. In: Proceedings of ECIR 2007, Rome, Italy (2007)
16. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of EMNLP 2011
17. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::similarity—measuring the relatedness of concept. In: Proceedings of 5th NAACL, Boston, MA (2004)
18. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Commun. ACM 18 (1975)
19. Landauer, T., Dumais, S.: A solution to plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychol. Rev. **104** (1997)
20. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: *SEM 2013 shared task: semantic textual similarity, including a pilot on typed-similarity. In: *SEM 2013 (2013)
21. Schütze, H., Pedersen, J.O.: Information retrieval based on word senses. In: Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (1995)
22. Aston, G., Burnard, L.: The BNC Handbook: Exploring the British National Corpus with SARA. Edinburgh University Press, Scotland (1998)
23. Graff, D.: English Gigaword. Technical report, Linguistic Data Consortium, Philadelphia (2003)

24. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. LRE **43**(3), 209–226 (2009)
25. Schütze, H.: Word space. In: Advances in Neural Information Processing Systems 5, Morgan Kaufmann, pp. 895–902 (1993)
26. Basili, R., Pennacchiotti, M.: Distributional lexical semantics: toward uniform representation paradigms for advanced acquisition and processing tasks. Nat. Lang. Eng. **16**(4), 347–358 (2010)
27. Lin, D.: Automatic retrieval and clustering of similar words. In: COLING-ACL, pp. 768–774 (1998)
28. Fano, R.M., Hawkins, D.: Transmission of information: a statistical theory of communications. Am. J. Phys. **29**(11), 793–794 (1961)
29. Bengio, Y., Delalleau, O., Roux, N.L.: The curse of dimensionality for local kernel machines. Technical report, Departement d'Informatique et Recherche Operationnelle (2005)
30. Lee, J., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer, New York (2007)
31. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. J. Soc. Ind. Appl. Math.: Ser. B, Numer. Anal.
32. Johansson, R., Nugues, P.: Dependency-based syntactic-semantic analysis with PropBank and NomBank. In: Proceedings of CoNLL, pp. 183–187 (2008)
33. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
34. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS), pp. 625–632 (2001)
35. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: ECML, Machine Learning: ECML, Berlin, Germany, pp. 318–329 (2006)
36. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proceedings of NAACL, Stroudsburg, PA, USA, pp. 57–60 (2006)
37. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. In: Brodley, C., Danyluk, A. (eds.) Proceedings of ICML-01 18th International Conference on Machine Learning, Williams College, US, Morgan Kaufmann Publishers, San Francisco, USA, pp. 66–73 (2001)
38. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of ACL'02 (2002)

Evaluating Italian Parsing Across Syntactic Formalisms and Annotation Schemes

Anita Alicante, Cristina Bosco, Anna Corazza and Alberto Lavelli

Abstract This paper describes some results about the way syntactic representations and parsing methodologies affect the performance of systems for parsing Italian. Italian has a rich morphology, especially with respect to Verbal suffixes, that can provide a parser with useful information for making the correct choices. With respect to syntactic representation, the experiments are based on a treebank for Italian, which has been delivered both in a dependency and in a constituency formalism, and for each of them also annotated at different degrees of specificity. The two paradigms are compared, and the different degrees of specificity in marking some syntactic phenomena are pointed out. On the basis of this treebank, statistical parsers have been evaluated. The results have shown that both the representation format and the parsing approach strongly affect the performance, that in some cases are very close and in others drastically different from the ones that constitute the state of the art for English.

Keywords Parsing · Word order · Morphologically rich languages

1 Introduction

Whenever a data-driven approach to parsing is adopted, performance depends not only on the parser, but also on the training data, and on their specific characteristics. This is particularly true for statistical supervised approaches, which require annotated

A. Alicante (✉) · A. Corazza
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione,
Università di Napoli Federico II, Naples, Italy
e-mail: anita.alicante@unina.it

A. Corazza
e-mail: anna.corazza@unina.it

C. Bosco
Dipartimento di Informatica, Università di Torino,
C.so Svizzera 185, 10149 Turin, TO, Italy
e-mail: bosco@di.unito.it

A. Lavelli
HLT Research Unit, Fondazione Bruno Kessler, Povo, TN, Italy
e-mail: lavelli@fbk.eu

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_7

training sets. Developing such annotated data sets includes several aspects, which can be split in two main classes: the selection of the data and the annotation design.

A well-known bottleneck of real world data sets concerns their dimension, as annotated data are expensive and difficult to obtain. Moreover, the dimension of the data set can significantly influence the performance, together with the quality of the annotated material, measured e.g. in terms of larger variety of linguistic information made available by the annotation. When designing the annotation, a crucial point is in fact the choice of its level of detail. On the one hand, a less detailed or too generic annotation can be not suitable for conveying enough information for the considered application. On the other hand, an annotation that is too detailed would require a much larger training set to avoid data sparseness. In fact, when statistical approaches are adopted, it is not only necessary that all phenomena are represented, but also that their occurrences adequately represent their probability distribution.

In other works, e.g. [19], criteria to evaluate the data with respect to the parsing model have been proposed which are based on information theory. In this work, we try to find an optimal tradeoff between annotation precision and design, and the need of a large annotated training set, by considering parsing performance and by adopting two syntactic frameworks, namely constituency and dependency parsing (see respectively Sects. 4.1 and 4.2).

In addition to the annotation design, also the choice of texts to be included in the data set can be crucial for the final system performance. The more intuitive aspect regards the domain of the training texts. Also for domain influence we face a tradeoff similar in some way to the one we found for the choice of the annotation level, described in Sect. 5.

However, we aim also at considering adaptation to linguistic characteristics. In this work, we are considering an Italian data set, the Turin University Treebank (henceforth TUT, see Sect. 2 for a detailed description of the resource and of the features of the Italian language). A characteristic of Italian, in particular when compared to English, is that words can follow a freer order in constituent positioning. The last set of experiments presented in this paper is especially focussed on word order, described in Sect. 6. We will set apart the most frequent Italian construction, namely the case where the subject precedes the verb and the object, and we will therefore show how parsing performance varies when a training set focusing on this kind of construction is considered.

All in all, in this work we aim at designing an assessment procedure for treebank annotation which is based on parsing performance. The assessment protocol we worked out aims at contributing at two main research issues related to the design of annotation schemes and frameworks for treebanks.

First, we would like to investigate how different amounts of linguistic information annotated in a treebank can influence the results of parsing systems, both in dependency and constituency-based annotation. For what concerns dependency, we have therefore tested a parsing system on three different settings of grammatical relations that include more or less specific relations, which can be extracted from the TUT native format. For what concerns instead constituency, we have tested the parser on two different formats, namely TUT-Penn and Augmented-PEnn

(henceforth APE), where the latter is a Penn format enriched with dependency relations coming from TUT.

Second, we would like to address the following question: When we need a data-driven parsing system specialized on a particular task, is it more effective to build a smaller training set specialized on the task, or a larger and more general one? In particular, we consider two kinds of specialization: the former directed to different application domains, the latter to a specific linguistic characteristic, that is the word order.

In the next section we present the data sets employed in the experiments, by focusing the attention on the different types of annotation. Sect. 3 is then devoted to the discussion of the data-driven parsers employed in the experiments. In Sects. 4, 5, and 6, we present three sets of experiments. The first one aims at finding the best annotation scheme which can be adopted with the considered data set. Then we focus our attention on the domain dependence of the data, and, in Sect. 6 we consider the dependence on the kind of syntactic construction which is prevalent in the training data. Some final remarks and proposals for future work conclude the paper.

2 The Italian Treebank TUT

The Turin University Treebank is a resource developed in the last 10 years by the Natural Language Processing group of the University of Turin [2–4, 10, 12]. The core of the treebank is a dependency-based annotation scheme centered on the predicate-argument structure, and a rich representation of morphological features, as needed for Italian (described in Sect. 2.1). Moreover, the resource has been enriched by the converted versions in a Penn-like format ([5, 11], described in Sect. 2.2), and in a CCG format [7].

The data of TUT are currently organized in six corpora according to text genre,[1] as shown in Table 1, and they consist in 102,150 tokens[2] in TUT native format, which correspond to 84,666 words, 10,056 punctuation marks and 7,428 null elements. The treebank annotation is mainly performed by the TULE parsing system [27–30], a rule-based parser which has been also applied to English, Catalan and French. The output of this rule-based parser is semi-automatically checked and manually corrected to produce the gold standard of the treebank.

The annotation schemes and formats of TUT have been designed mainly taking into account the features of Italian which are typical of Morphologically Rich Languages (MRLs), i.e. rich inflection, pro-drop, free word order and discontinuity, amalgams. The richness of the inflection strongly impacts on the design of the

[1] The CODICECIVILE and COSTITA corpora include legal texts, the EUDIR declarations of the European Community from the Italian section of the JRC-Acquis Multilingual Parallel Corpus (see http://langtech.jrc.it/JRC-Acquis.html). Instead NEWS corpus includes texts from Italian newspapers, WIKIPEDIA from the Italian section of Wikipedia, and VED a miscellanea from academic, journal and novels.

[2] The term token refers to all the objects annotated in the treebank, namely words, punctuation marks and null elements.

Table 1 The composition of TUT subcorpora in terms of number of sentences, words per sentence, punctuation marks per sentence, null elements per sentences (including also pro-drop subjects), pro-drop subjects per sentence, and amalgams per sentence

Corpus	Sentences	Words	Puncts[a]	Null	Pro-drops	Amalgams
CODICECIVILE	1,100	25.50	3.08	2.40	0.21	2.13
NEWS	700	25.78	2.58	1.56	0.22	1.85
VED	400	33.14	4.02	2.84	0.64	1.91
EUDIR	201	37.09	3.50	2.54	0.28	3.58
WIKIPEDIA	459	32.13	2.86	2.22	0.34	2.39
COSTITA[b]	682	19.32	1.82	1.51	0.17	1.74
All	3,542	26.74	2.84	2.09	0.28	2.09

[a] Because punctuation marks are annotated in TUT, they are included as words in these counts
[b] The annotation of the corpus of the Costituzione Italiana is developed within the PARLI project partially funded by the Italian Ministery of the Research and Instruction

Part of Speech (PoS) tag set for the dependency native TUT format; whilst to make adequate for Italian the Penn format (i.e. TUT-Penn, as described in [4, 9, 13]) a more fine-grained representation of Verb tenses has been adopted with respect to the one adopted for English in the Penn Treebank. In order to explicitly represent the argument structure of each Verb, a trace-filler mechanism has been applied also in the dependency-based representation of TUT thus giving information useful for the identification of the subject and main complements even when not lexically realized, e.g. pro-drop. The same representational tool is applied for the annotation of non-projective structures. For the tokenization of the amalgamated words, among the different possible strategies, we assumed an explicit representation of each of their parts as separated morpho-syntactic items.

In the following sections, the dependency and constituency formats of TUT are presented in more details.

2.1 Dependency-Based Annotation in TUT

TUT native scheme for the dependency-based annotation is featured by two main characteristics. On the one hand, the structure which is mainly based on a theoretical framework for dependency grammar, but also made more adequate for the reference language with the necessary representational tools (like the null elements, see below) and choices (see e.g. the asymmetric representation adopted for coordination). On the other hand, the grammatical relations that label the tree edges of the treebank. Moreover, as mentioned before, a rich tag set has been used for the PoS annotation, which includes 16 grammatical categories further specialized by 43 types associated to several features.

For what concerns the structure, a typical TUT tree shows a pure dependency format centered upon the notion of argument structure and applying the major principles of the *Word Grammar* theoretical framework [23]. This is mirrored, for instance, in the annotation of Determiners and Prepositions which are represented in TUT trees

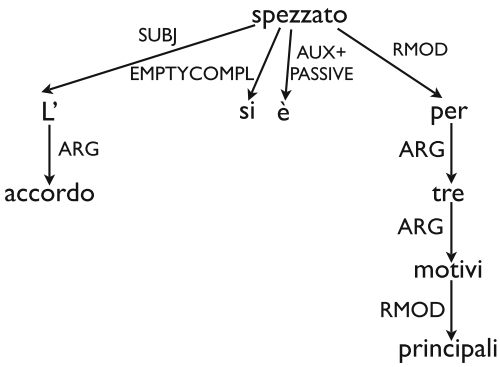

Fig. 1 Sentence NEWS-355 in 1-Comp setting

as complementizers of Nouns or Verbs. For instance, in Fig. 1 the tree for the sentence NEWS-355 from TUT, i.e. "*L'accordo si è spezzato per tre motivi principali*" (The agreement has been broken for three main motivations),[3] shows the features of the annotation schema. In particular, we see the role of complementizer played by Determiners (i.e. the article "*L*"' (The) and the numeral "*tre*" (three)) and Prepositions (i.e. "*per*" (for)), and the selection of the main Verb as head of the structure instead of the auxiliary. According to the Word Grammar, since the classes of Determiners and Prepositions include elements[4] which often are used without complements and can occur alone (like possessive and deictic Adjectives or Numerals used as Pronouns, or Prepositions like 'before' and 'after'), all the members of these classes play in TUT trees the same head role when they occur with or without Nouns or Verbs.

By contrast, the TUT scheme exploits also representational tools which are non-standard in dependency-based annotations, i.e. null elements, in order to deal with structures that are challenging for dependency-based formats or to give information crucial for some task.[5]

Null elements are used in the representation of long distance dependencies and elliptical structures, and, in general, to make explicit Verb complements when they are not lexically realized, namely in the case of equi.[6] and pro-drop phenomena. By contrast with other treebanks where null elements are not used, in all these cases,

[3] English translations of the Italian examples are literal and so may appear awkward in English.

[4] According to the Word Grammar, many words qualify as Prepositions or Determiners which traditional grammar would have classified as AdVerbs or subordinating conjunctions.

[5] For instance, in Machine Translation if the source language allows argument deletion and the target language does not, in order to make possible for the system to handle the translation, it is crucial that in the source language the dropped argument is explicitly marked. An alike situation can happen in a translation from Italian (a typical pro-drop language where the subject deletion is very common with tensed Verbs) to English (where the subject is always lexically realized in tensed clauses).

[6] The term equi refers to the lacking Subject of the subordinate infinitive Verb, e.g. the Subject of the Verb "dormire" (sleep) in "Vuole dormire" ([He] wants [to] sleep).

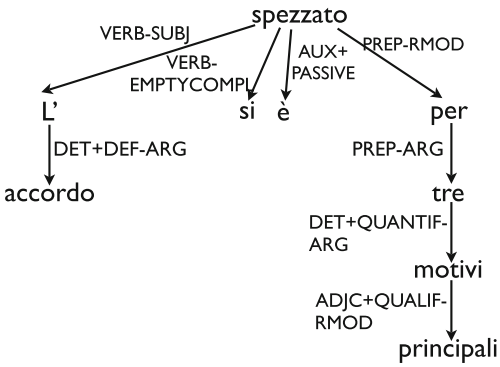

Fig. 2 Sentence NEWS-355 in 2-Comp setting

null elements permit dependency trees to be without crossing edges, but they allow also for the recovery of projective structures for sentences which are not projective.

Nevertheless, it should be observed that, in order to be compliant with the standard adopted in the evaluation procedures, e.g. in parsing contests, the format used in the experiments reported in this paper is the one adopted in the CoNLL shared tasks in 2006 and 2007, where null elements are not allowed.[7]

For what concerns, instead, the grammatical relations that label the tree edges, TUT exploits a rich set of grammatical relations designed to represent a variety of linguistic information according to three different perspectives, i.e. morphology, functional syntax and semantics. The main idea is that a single layer, the one describing the relations between words, can represent linguistic knowledge that is proximate to semantics and underlies syntax and morphology, which seems to be unavoidable for efficient processing of human languages, i.e. the predicate argument structure of events and states. Therefore, each relation label can in principle include three components, i.e. morpho-syntactic, functional-syntactic and syntactic-semantic, but can be made more or less specialized, including from only one (i.e. the functional-syntactic) to three of them. For instance, the relation used for the annotation of locative Prepositional modifiers, i.e. PREP-RMOD-LOC (which includes all the three components), can be reduced to PREP-RMOD (which includes only the first two components) or to RMOD (which includes only the functional-syntactic component).

This works as a means for the annotators to represent different layers of confidence in the annotation, but can also be applied to increase the comparability of TUT with other existing resources, by exploiting the amount of linguistic information more adequate for the comparison, e.g. in terms of number of relations. Since in different settings several relations can be merged in a single one (e.g. PREP-RMOD-TIME and PREP-RMOD-LOC are merged in RMOD), each setting includes a different number of relations: the setting based on the single functional-syntactic component (henceforth *1-Comp*) includes 72 relations, the one based on morpho-syntactic and functional-syntactic components (*2-Comp*) 140, and the one based on all the three components (*3-Comp*) 323. If we compare the tree in Fig. 1, with the trees in Figs. 2

[7] The projectivity constraint is maintained for TUT also in the CoNLL format.

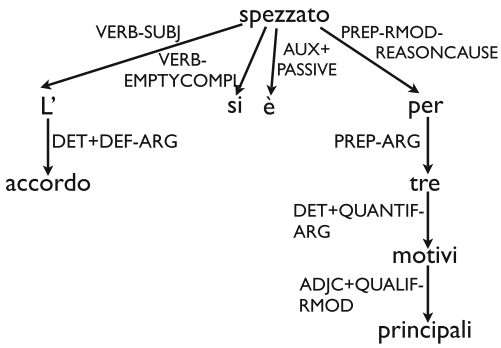

Fig. 3 Sentence NEWS-355 in 3-Comp setting

and 3, we see also the variation of relations in the three settings for the same sentence. For instance, the relation between *spezzato* (broken) and the Prepositional modifier *per tre motivi principali* (for three main motivations), or the argument articles that are ARG in 1-Comp and DET+DEF-ARG (i.e. ARGument of a DEFinite Determiner) in the other settings. The last case is an example of relation that does not include semantic information and therefore remains the same in 2- and 3-Comp settings.

2.2 Constituency-Based Annotation in TUT

Beyond the above described dependency-based format, TUT features also some constituency-based annotation. All these formats are derived by native TUT through an automatic conversion whose final output is a Penn format customized for Italian, i.e. TUT-Penn [5, 6, 11, 14]. A methodology that consists in organizing the conversion in steps to be performed in cascade has been in fact applied, and a set of parallel annotations has been generated as a side effect of the conversion in TUT-Penn itself. Each step of this process outputs in practice a new format, which differentiates from the input one only with regard to a single kind of knowledge, such as morphological (e.g. PoS tag set conversion), structural syntactic (e.g. conversion from dependency to constituency), functional syntactic (e.g. conversion of grammatical relations), and a separate analysis of each kind of knowledge is in this way implemented. Moreover, starting from one of these formats intermediate from TUT to TUT-Penn, the CCG-TUT, a treebank of Combinatory Categorial Grammar derivations [7] has been developed.

In the rest of this section, we will mainly focus on the TUT-Penn format, for the others all the details can be found in the above mentioned references.

The Penn Treebank format as described in the guidelines for English generalizes well to other languages, both to languages with very little inflectional morphology, like Chinese,[8] or richer, like Arabic[9] [21]. But each language can contain linguistic

[8] See http://www.cis.upenn.edu/chinese/.

[9] See http://www.ircs.upenn.edu/arabic/.

phenomena unseen in English that can be addressed in some way when Penn format is applied to it. For what concerns Italian, the main differences with respect to English refer to the PoS tag set, as expected for a MRL, and the word order, which is more free in Italian like in other MRLs. We will focus on these topics in the rest of this section.

The PoS tags associated to terminal nodes in constituency-based treebanks vary to a large extent according to the specific language of the corpus, and the cardinality of tag sets in use clearly reflects differences among languages in terms of inflectional richness. For instance, as reported in [18], where a partial conversion of the Prague Dependency Treebank (PDT [22]) in Penn is developed, the tag set of the English Penn Treebank is poorer than the one for Czech of the PDT. Nevertheless, very large PoS tag sets can lead to serious sparse data problems and make results of parsers and taggers trained on data annotated with those tag sets not comparable with the results obtained on the Penn Treebank. Therefore, some form of reduction of the PoS tag set is usually operated in too large PoS tag sets for the conversion in Penn format, see e.g. [18] and http://www.coli.uni-sb.de/sfb378/negra-corpus/.

For Italian, as for other MRLs, a rich morphological tag set is applied in the native TUT: it includes 16 grammatical categories further specialized by 43 types, which are associated with a large variety of features. By contrast, in the TUT-Penn, following the experiences of the conversion of NEGRA Treebank and PDT, the tag set has been reduced to 68 tags only, versus 36 in the English Penn Treebank. Beyond the information that Penn tag set makes explicit,[10] TUT-Penn takes into account a richer variety of features for Verbs, Adjective and Pronouns. For instance, verbal PoS tags show fine-grained temporal information and distinguish among three classes of Verbs (Modal, Auxiliary, Main), rather than two in Penn (Modal and non-Modal), thus making explicit the different potentiality of those classes with respect to the representation of predicative argument structures. Distinctions drawn in the tag set among various types of Adjectives and Pronouns enable instead the recovery of information such as the owner of an object (possessive Adjective) or the referent of a Pronoun as a location (locative Pronoun). By contrast, by following the findings presented in [18],[11] the features concerning person, gender and number are not included in the TUT-Penn tag set (except in the case of Nouns where the number is annotated according to Penn).

Moreover, for sake of conciseness typical of the Penn format, the format of TUT-Penn includes PoS tags in a Penn-like compact version where each morphological feature of a single word is expressed by a few letters and encompassed with the other features of the same word in a short single string, as in the following example.[12]

[10] Apart from a few cases of English morphological features which do not exist (e.g. possessive ending) or do not correspond with Italian forms (e.g. comparative Adjective and Adverb).

[11] The inclusion of person, gender and number values in morphological tags were tested without yielding any improvement in the parser performance. The investigation of the effect of the inclusion of these features in the Italian case, or in that of other MRLs, can be of some interest for future works.

[12] English translation: The agreement is broken for three main motivations.

```
( (S
  (NP-SBJ (ART~DE L') (NOU~CS accordo))
  (NP (PRO~RI si))
  (VP (VAU~RE é)
      (VP (VMA~PA spezzato)
          (PP (PREP per)
              (NP (NUMR tre) (NOU~CP motivi) (ADJ~QU principali)))))
  (. .)) )
```

For instance, the compact tag for a *NOun Common SIngular* is *NO~CSI*, that for a *Noun Common PLural NO~CPL*, that for a *Noun Proper NO~P*.[13]

Among the PoS tags used in Penn, 22 are basic and 14 represent additional morphological features, while in TUT 19 are basic and 26 represent features. In both the tag sets, each feature can be composed with a limited number of basic tags, e.g. in TUT-Penn *DE* (for demonstrative) can be composed only with the basic tags *ADJ* (for adjective) and *PRO* (for pronoun). For Verbs, in Penn a Verb can be annotated by using the basic tag *VB* (for basic form) or with a feature, e.g. *VBD* (for past tense) or *VBG* (for gerund or past participle). In TUT-Penn, the verb is instead annotated as *VMA* (for verb main), *VMO* (for modal) and *VAU* (for auxiliary) always associated with one of the 11 features that represent the conjugation, e.g. *VMO~PA* for a modal verb in participle past or *VMA~IM* for a main verb in imperfect.

In TUT-Penn null elements are used like in Penn, but also to deal with some typical features of MRLs, e.g. for marking Subjects which occur in non standard position with respect to the Verb or pro-drops. While the position of the Subject with respect to the Verb is not an issue in formats where the relation Subject is not structurally marked, like in dependency,[14] this is a problem in constituency-based representations where the Subject is known as *external argument* of the Verb and is assumed to be in a pre-verbal position and is not included in the same phrase of the Verb (i.e. VP) like the other verbal arguments. The following example[15] shows how TUT-Penn deals with this phenomenon.

```
( (S
  (NP-SBJ (-NONE- *-533))
  (PP (PREP Al)
    (NP (ART~DE Al) (NOU~CS proprietario)))
  (VP (VAU~RE é)
    (VP (VMA~PA dovuta)
      (NP-EXTPSBJ-533 (ART~IN una) (ADJ~QU giusta) (NOU~CA indennità))))
  (. .)) )
```

In this example, i.e. CODICECIVILE-40, it can be seen the use of the special functional tag *EXTPSBJ* for the annotation of the Subject in post-verbal position and the null element co-indexed with this Subject which is positioned in the canonical position of the Subject. The same annotation for null elements and co-indexing as in Penn is adopted in TUT-Penn.

---

[13] Proper nouns are not marked in Italian in terms of number.

[14] In fact, in a dependency tree the relation subject marks an edge linking the verbal head with a dependent which can be distinguished from other verbal dependents only by the type of the relation.

[15] English translation: A right allowance is due to the owner.

For what concerns relations, it is known that Penn assumes a limited set of them, namely a few of functional tags which can be associated with phrases. In TUT-Penn the same inventory of functional tags is adopted with the exception of some that are very specific and not useful in languages other than English.[16]

Nevertheless, to expand the possibility of cross-framework and cross-paradigm comparison, assuming the importance of the representation of the predicate argument structure, also in a constituency based representation and for a variety of tasks, we developed as a step of the conversion process from TUT to TUT-Penn, a format that structurally corresponds to Penn but maintains, where possible, the functional-syntactic knowledge encoded in the native dependency TUT, i.e. APE. This format is applied in the following example, where the Penn structure is enriched with the functional labels EMPTYCOMPL, RMOD-REASONCAUSE, ARG and PUNCT-END.

```
( (S
  (NP-SBJ (ART~DE L') (NOU~CS accordo))
  (NP-EMPTYCOMPL (PRO~RI si))
  (VP (VAU~RE é)
      (VP (VMA~PA spezzato)
          (PP-RMOD-REASONCAUSE (PREP per)
              (NP-ARG (NUMR tre) (NOU~CP motivi) (ADJ~QU principali)))))
  (PUNCT-END .)) )
```

3 Parsers Employed in the Assessment

All the parsing experiments are performed on the TUT data set discussed in Sect. 2 by using the two statistical parsers discussed in this section, namely the Berkeley parser for the constituency model and MaltParser for the dependency one. Indeed, these two parsers have shown state-of-the-art performance during EVALITA 2009 [14, 15].

The Berkeley parser [33] is a constituency parser based on a hierarchical coarse-to-fine parsing, where a sequence of grammars is considered, each being the refinement, namely a partial splitting, of the preceding one. Its performance represents the state of the art for English and for other languages. An interesting characteristic is that porting the Berkeley parser to a new language requires no additional effort apart from the availability of a treebank. Constituency parser performance is evaluated as usual by labeled precision (LP) and recall (LR) and F_1. In the experiments presented in this section we performed 5 iterations.

MaltParser [32] is a data-driven dependency parser that was one of the top performing systems in the multilingual track of the CoNLL shared tasks on dependency parsing in 2006 and 2007 and in the Evalita 2009 dependency Parsing Task (henceforth EPT) for Italian. Dependency parser performance is evaluated in terms of Labeled Attachment Score (LAS).

[16] E.g. the tag *PUT* which represents the locative complement of the Verb "put", or the tag *DTV* (dative) which is annotated in indirect objects when they are realized as prepositional phrases, i.e. not affected by the dative shift.

4 Comparing Different Annotations

As already discussed, instead of looking for a theoretical optimal trade-off between annotation detail and parsing accuracy, we deal with the problem experimentally. Therefore we have performed experiments to find the annotation detail leading to the best parsing performance. However, the conclusions which can be drawn from such experiments are valid only in the same conditions in which the tests have been conducted. In particular, we need to distinguish two cases depending on the considered paradigms, which can be either constituency or dependency.

4.1 Constituency

In the constituency framework for parsing, the importance of punctuation is well recognized. While in the Penn format punctuation is marked by the pair token-PoS, in the APE three different tags are adopted for punctuation, namely `PUNCT-END`, `PUNCT-SEPARATOR` and `PUNCT-COORD`. In all the experiments we describe in the following sections, we always maintain such distinction.

The **baseline system** uses the TUT-Penn format of the considered dataset (see Sect. 2). Therefore we consider the following seven annotation levels:

L1 **Specific**. Only the first chunk of all suffixes is conserved together with the label prefix, for example the `NP-RMOD-LISTPOS` becomes `NP-RMOD`. All substrings which follow the characters + or ~ or the second instance of − are deleted.

L2 **Most Generic**. Delete all suffixes, for example `NP-RMOD-LISTPOS` becomes `NP`.

L3 **Frequent nonterminal and PoS labels**. In this strategy, in addition to all punctuation tags, we only maintain the suffixes which contain all PoS tags and nonterminal labels having more than 1,000 occurrences, which are reported in Table 2.

Table 2 List of nonterminal and PoS labels with more than 1,000 occurrences

PoS		Nonterminal	
ART~DE	7,642	NP-ARG	8,341
NOU~CS	7,424	PP-RMOD	3,343
NOU~CP	3,311	NP-OBJ	1,941
ADJ~QU	2,958	S-ARG	1,385
VMA~PA	1,866		
VMA~RE	1,511		
NOU~PR	1,322		
VMA~IN	1,304		

L4 **Delete everything except punctuation**. In this case we preserve only all punctuation suffixes, that is `PUNCT-*`, deleting the others.

L5 **Frequent nonterminal labels**. In this strategy we only maintain the punctuation tags (`PUNCT-END`, `PUNCT-SEPARATOR` and `PUNCT-COORD`) and the suffixes which contain non terminal labels occurring more than 1,000 times, reported in the second column of Table 2.

L6 **Frequent (>1,000) PoS labels**. In this strategy we only preserve all punctuation tags, that is `PUNCT-*` and suffixes which contain the PoS labels occurring more than 1,000 times, reported in the first column of Table 2. The other suffixes have been deleted.

L7 **NP-SUBJ and NP-OBJ**. All additional annotation is deleted with the only exception of punctuation and two nonterminal labels, namely `NP-SUBJ` and `NP-OBJ`, that allow the distinction of cases where noun phrases assume the role respectively of subject and direct object.

In order to best exploit the data, we adopt an N-fold cross-validation experimental protocol. Whenever not otherwise specified, we choose $N = 10$. In addition to the usual Parseval metrics, namely Labelled Recall (LR), Labelled Precision (LP) and F-measure (F_1), we also consider the Exact Matching Rate (EMR), i.e. the rate of parse trees which are completely correct (Table 3).

The performance of all strategies is worse than the baseline. The best choice among all the variations is the one keeping only the most frequent tags. Furthermore, note that the most generic annotation performs better than the most specific one. This is probably due to data sparseness, as we need much more data to collect sufficient statistics for a more detailed annotation. Another important conclusion can be drawn from the relatively good performance of the experiments keeping only punctuation: indeed, it results evident that punctuation includes most of the information necessary to correctly reconstruct the skeleton of the analysis of the sentences. In fact, in these experiments no other label than punctuation is considered. In addition to that, the more generic annotation labels can be identified more easily than the more detailed ones.

All in all, we can conclude that the most important cue to build the skeleton of analyses is represented by punctuation. After that, PoS tags are the second crucial factor to direct the analysis of sentences. An explanation for this could be that the PoS tags depend directly on the input sentence, and can therefore be determined with little or no error.

4.2 Dependency

In the dependency framework we considered the influence on evaluation scores of the language, the frequency of hard to parse constructions, and mainly the design of the annotation schema [8]. Our analysis is based on TUT and MaltParser, and the experiments focussed on a set of Italian hard to parse constructions and the three

Table 3 Comparison among the different strategies by using a 10-fold cross validation protocol

	LR	LP	F_1	EMR
Baseline				
Berkeley-iteration #5	78.06	78.63	78.35	25.85
Berkeley-iteration #5 $\leq$ 40	81.49	81.83	81.66	31.24
Specific				
Berkeley-iteration #5	61.15	62.04	61.59	13.25
Berkeley-iteration #5 $\leq$ 40	66.01	64.83	65.42	15.99
Most Generic				
Berkeley-iteration #5	76.31	76.52	76.52	25.84
Berkeley-iteration #5 $\leq$ 40	79.91	80.15	80.03	31.18
Frequent nonterminal and PoS labels				
Berkeley-iteration #5	68.81	68.53	68.67	16.17
Berkeley-iteration #5 $\leq$ 40	72.49	71.99	72.24	19.66
Delete everything except puncutation				
Berkeley-iteration #5	77.09	76.88	76.99	25.35
Berkeley-iteration #5 $\leq$ 40	80.71	80.68	80.69	30.57
Frequent nonterminal labels				
Berkeley-iteration #5	69.84	70.12	69.98	17.52
Berkeley-iteration #5 $\leq$ 40	73.71	73.76	73.73	21.23
Frequent (>1,000) PoS tags				
Berkeley-iteration #5	77.55	77.50	**77.52**	24.76
Berkeley-iteration #5 $\leq$ 40	80.86	80.60	**80.73**	29.98
NP-SUBJ and NP-OBJ				
Berkeley-iteration #5	73.25	73.46	73.35	18.62
Berkeley-iteration #5 $\leq$ 40	76.63	76.53	76.58	22.62

The best performance is marked in bold

settings of the annotation schema of TUT, i.e. 1-, 2- and 3-Comp (see Sect. 2.1), which vary with respect to the amount of underlying linguistic information.

The approach we propose is language oriented and construction-based, but it differs e.g. from those in [25, 34]. In particular, the selection of the hard to parse phenomena for our experiments is motivated not only by linguistic and applicative considerations, as in these related works, but also driven by the performance of different parsers. Assuming that most of the parsing errors are related to some specific relation and construction, first of all we identify cases that can be considered as hard to parse for Italian by analyzing and comparing the results of the six participant parsers at the EPT [15]. The test set on which the parsers were applied included 240 sentences (5,287 tokens) balanced alike to those of the treebank used for training: 100 sentences (1,782 tokens) from newspapers, 100 (2,293 tokens) from Civil Law Code and 40 (1,212 tokens) from the Passage/JRC-Acquis corpus. We compute precision

and recall[17] for each type of grammatical relations. To further assess the results, we perform the same kind of evaluation on the three relation settings running a 10-fold cross validation on the entire treebank with MaltParser. After identifying the hard to parse relations, we develop a comparative analysis of the behavior of MaltParser in such cases.

We identify the following hard to parse constructions:

- the predicative complement of the object, i.e. PREDCOMPL+OBJ (which occurs 141 times in the full treebank, i.e. 0.19 %). For instance, in "*Il parlamentare si è detto **favorevole** ad una maggiore apertura delle frontiere ai rifugiati politici.*" (The parliamentarian itself has said **in favour** of a major opening of frontiers to the political refugees.)
- the indirect object, i.e. INDOBJ (which occurs 325 times, i.e. 0.45 %). For instance, in "*Noi non permetteremo **a nessuno** di imbrogliarci.*" (We will not allow **to anybody** to cheat us.)
- various relations involved in coordinative structures that represent comparisons (e.g. COORDANTEC+COMPAR and COORD+COMPAR (which occurs 64 times, i.e. 0,08 %), like in "*Usa un test **meno** raffinato **di quello tradizionale***." ([He] exploits a test **less** refined **than the traditional one**.))
- various relations for the annotation of punctuation, in particular SEPARATOR, OPEN+PARENTHETICAL (which occurs 1,116 times, i.e. 1.5 %) and CLOSE+PARENTHETICAL (which occurs 1097 times, i.e. 1.5 %)). For instance, SEPARATOR (which occurs 1,952 times, i.e. 2.7 %) is used in cases where commas play the role of disambiguating marks and an ambiguity could result if the marks were not there [24], e.g. in "*Quando il meccanismo si inceppa, è il disastro.*" (When the mechanism hinds itself, is a disaster). OPEN+/CLOSE+PARENTHETICAL are instead used for the annotation of paired punctuation that marks the parenthetical in "*Pochi quotidiani**,** solo quelli inglesi**,** saranno oggi in vendita.*" (Few newspapers**,** only those English**,** will be today on sale.)

Since not all the grammatical relations of 1-Comp occur in the test set, the above list cannot in principle be considered as representative of how hard to parse is the treebank (and the Italian language). A 10-fold cross validation performed on the whole TUT with the 1-Comp setting shows that other low-scored relations exist, but since they appear with a very low frequency we did not include them in our experiments.[18] The comparison with ISST-TANL, developed in [15, 16], shows that similar relations are low-scored also in this other resource, notwithstanding the different underlying annotation schema.

First of all, we analyze the distribution of hard to parse relations and constructions in the data. To obtain the following results we exploit as the experimental protocol the 10-fold cross validation. The application of MaltParser on the treebank with the 1-Comp setting shows that the performance significantly varies when the parser is

[17] The evaluation has been performed by using the MaltEval tools [31].

[18] This shows however that the test set, even if it shows the same balancement of TUT, does not represent at best the treebank in terms of relations and constructions.

Table 4 MaltParser scores in 10-fold cross validation over the whole treebank

	1-Comp	2-Comp	3-Comp
LAS	83.24	82.56	78.77
UAS	87.69	87.60	87.20

Table 5 MaltParser scores for COORD+COMPAR with different settings

	EPT	1-Comp	2-Comp	3-Comp
Prec	50.00	89.66	83.33	86.21
Rec	25.00	54.17	52.08	52.08

Table 6 MaltParser scores for (VERB-)PREDCOMPL+OBJ with different settings

	EPT	1-Comp	2-Comp	3-Comp
Prec	50	57.81	60.00	61.16
Rec	40	52.48	53.19	52.48

applied to the EPT test set rather than to all the treebank, i.e. from LAS 86.5 and UAS 90.96, in the test set [26], to LAS 83.24 e UAS 87.69 in all TUT.[19] This suggests that the distribution of hard to parse phenomena is not the same in both cases.

Second, in order to test the hypothesis that the degree of difficulty of the same hard to parse constructions can vary in the test set with respect to the treebank, we first analyze the performance of MaltParser on all TUT with the 3 settings, and, second, we analyze the variation of precision and recall for each hard to parse case according to the three settings. As Table 4 shows, the performance in terms of UAS is not significantly influenced by the different settings, since the difference concerns the relation labels rather than the tree structures. Instead, LAS decreases when the number of relations is enlarged in settings that should be more informative, going from 72 (1-Comp), to 140 (2-Comp), to 323 relations (3-Comp). The larger amount of relations occurring a small number of times in 2- and 3-Comp (with respect to 1-Comp) increases the sparseness of relations and negatively influences the performance. Also the stability across all settings of the performance only on more frequent relations, further supports this conclusion.

Finally we focus on single hard to parse relations in order to show the variation of parser performance in the three settings. Tables 5, 6 and 7 show that the parser behavior varies in a different way for different relations and sometimes following a different trend with respect to the results on all the treebank. For instance, for COORD+COMPAR (Table 5) the best performance is in 1-Comp and the worst in the EPT test set. For PREDCOMPL+OBJ (Table 6), instead, the best performance is in 3-Comp and the worst in the EPT test set. Therefore, in this case there is a contrast

[19] This is only partially explained by the sentence length, which is lower than 40 words only in the test set, and by the smaller size of the training set for the 10-fold cross validation.

Table 7 MaltParser scores for (VERB-)INDOBJ with different settings

	EPT	1-Comp	2-Comp	3-Comp
Prec	68.97	57.00	55.96	48.26
Rec	58.82	52.35	50.49	63.19

Table 8 MaltParser scores on 1-, 2- and 3-Comp TUT with and without punctuation, in 10-fold cross validation

	1-Comp	2-Comp	3-Comp
LAS Punct	83.24	82.56	78.77
LAS noPunct	86.78	86.02	81.88
UAS Punct	87.69	87.60	87.20
UAS noPunct	91.10	91.01	90.70

with the general trend shown in Table 4, since the results are significantly better when the relation labels include the morphological component.

For what concerns instead punctuation, it should be noted that it is not always considered when evaluating parsing performance. As we have seen before, in our experimental evaluation punctuation is instead taken into account, but the related relations are among the low-scored ones. For instance, SEPARATOR is in the set of the 9 most frequent relations[20] (in 1-Comp setting in both all the treebank and the test set) and occurs around 2,000 times in the full treebank, but it is the one scoring the lower in precision and recall of this set for all the parsers participating to the EPT. Therefore, in the perspective of a comparison with other evaluations and resources, it would be useful to see how our results vary when punctuation is excluded, as in Table 8. The UAS and LAS scores of MaltParser are in all TUT settings 3.5 points higher when the punctuation is not taken into account. As for ISST-TANL, the experiments show that the difference in performance when considering or not considering punctuation is between 1.76 and 2.50 according to different parser parameters. This smaller difference can be at least in part motivated by the different approach adopted in ISST-TANL for the annotation of punctuation, which is less detailed and based on a single relation (i.e. PUNC). This means that some improvement in parsing can be obtained by more adequate processing of punctuation, as said e.g. in [17], and/or by more adequate annotation of it. In fact punctuation is often relevant from a linguistic point of view as a marker of clause or phrase boundaries, thus if a parser does not predict it correctly, it can lead to incorrect parses and lower scores when evaluated against a resource that annotates punctuation.

[20] The ten most frequent relations in all the 1-Comp treebank (with respect to 72,149 annotated tokens) are ARG (30.3 %), RMOD (19.2 %), OBJ (4.5 %), SUBJ (3.9 %), END (3.3 %), TOP (3.2 %), COORD2ND+BASE (3.1 %), COORD+BASE (3.1 %), SEPARATOR (2.7 %), INDCOMPL (1.9 %).

As for the comparison with other languages, we have seen that part of the hard to parse phenomena for Italian are included also in the test suites proposed for German, e.g. forms of coordination. But, since the lists presented in [25, 34] are mainly linguistically motivated and not quantitatively determined, we cannot go beyond this observation and further extend the comparison.

For what concerns single phenomena, following the idea that parsing can be made more or less hard by the availability of different amount of linguistic information, we have seen that different effects can be caused by the use of more or less informative grammatical relations. The results demonstrate, in particular, that the evaluation based on the test set is limited with respect to the distribution and type of hard to parse constructions, which in the test set and in the entire treebank can be different, and the degree of difficulty of hard to parse constructions, which in the test set and in the entire treebank can be not the same.

5 Domain Influence

Domain influence is a well known task related to parsing. As far as Italian is concerned, it has been included among the official tasks of the last edition of Evalita in 2011 [20].

As discussed in Sect. 2, the treebank is composed by various corpora that represent different text genres, which can be broadly categorized in two different domains, namely *civil law*[21] and *newspapers*. The sublanguage used in each of the two parts is likely to be different from the other. To verify this conjecture we performed a new set of tests by considering each part separately. More precisely, as the two parts contain exactly the same number of sentences, we applied a 5-fold cross validation on each of the two parts. We chose the annotation level L6 which obtained the best performance on the treebank considered as a whole (see Sect. 4.1). In this annotation, all the rare PoS labels are deleted.

However, while on the whole treebank we assumed to be "rare" all PoS tags having less than 1,000 occurrences, in this case, as the treebank has been split in two, we consider a threshold of 500. In this way, also the following PoS tags have been considered in addition to the ones in Table 2 (the number of occurrences has been reported in parentheses): ART˜IN (863), NOU˜CA (845), VAU˜RE (817), PRO˜RE (777), PRO˜RI (634), and PRO˜PE (535).

The results in Table 9 are more consistent as they are less likely to be prone to data sparsity. Furthermore, they show that the civil law domain is easier to parse than the newspaper one. Moreover, the average of the F_1 for the two different domains

[21] For what concerns in particular parsing of legal text, see also the Proceedings of the LREC 2012 Workshop on Semantic Processing of Legal Texts (SPLeT-2012), available at http://www.lrec-conf.org/proceedings/lrec2012/workshops/27.LREC%202012%20Workshop%20-Proceedings%20SPLeT.pdf.

Table 9 Results obtained on the two domains separately by 5-fold cross-validation; in the lower part of the table, the results on the two domains are merged together

	LR	LP	F_1	EMR
Civil law, baseline				
Berkeley-iteration #5	79.43	79.51	79.47	31.12
Berkeley-iteration #5 $\leq$ 40	83.09	82.94	83.01	37.21
Civil law, frequent (>1,000) *PoS*				
Berkeley-iteration #5	79.07	78.35	78.72	33.33
Berkeley-iteration #5 $\leq$ 40	83.30	82.63	82.97	39.71
Civil law, frequent (>500) *PoS*				
Berkeley-iteration #5	79.40	78.59	78.99	31.16
Berkeley-iteration #5 $\leq$ 40	83.08	82.19	82.63	37.11
Newspaper, baseline				
Berkeley-iteration #5	71.76	71.72	71.74	14.90
Berkeley-iteration #5 $\leq$ 40	75.52	75.33	75.43	18.31
Newspaper, frequent (>1,000) *PoS*				
Berkeley-iteration #5	71.76	70.85	71.30	14.18
Berkeley-iteration #5 $\leq$ 40	75.54	74.33	74.93	17.41
Newspaper, frequent (>500) *PoS*				
Berkeley-iteration #5	72.42	71.63	72.03	14.88
Berkeley-iteration #5 $\leq$ 40	75.89	74.88	75.38	18.26
All together, baseline				
Berkeley-iteration #5	75.49	75.50	75.50	23.01
Berkeley-iteration #5 $\leq$ 40	79.22	79.05	79.14	27.85
All together, frequent (>1,000) *PoS*				
Berkeley-iteration #5	75.34	74.52	74.93	23.81
Berkeley-iteration #5 $\leq$ 40	79.37	78.41	78.88	28.75
All together, frequent (>500) *PoS*				
Berkeley-iteration #5	75.82	75.03	75.43	23.03
Berkeley-iteration #5 $\leq$ 40	79.41	78.46	78.93	27.80

is 75.61 for all sentences, and 79.22 for sentences shorter than 40 are both slightly better than the one obtained by merging the two domains.

5.1 Penn Format

As already discussed, adopting a more precise annotation gives more information in training, but it requires more precision in testing. In other words, parsing with a more informative format is a task more difficult than with a less informative format.

Therefore we apply the results of the preceding experiments to the task of parsing the Penn format, assuming that the best strategy involves preserving all frequent PoS tags (with at least 500 occurrences) and that it is better to use a domain dependent training set. In addition to that, we also try keeping all suffixes as in the most specific strategy.

First of all, for the baseline we delete all suffixes, including punctuation, on both the training and the test set. In this way we obtain an annotation which is very similar to the standard Penn format. We then train the parser on a richer annotated training set, parse the test set to obtain that richer annotation, and eventually delete all suffixes from the trees constructed by the parser. In this way, the output can be directly compared with that obtained by the baseline.

As in the preceding experiments, we applied a 5-fold cross validation separately to each of the two domains. Also in this case, the more specific annotation can not be parsed with the best performance. On the contrary, if we only keep the most frequent PoS tags, then we obtain better performance. However, also in this case the baseline presents the best performance (Table 10).

6 Word Order Influence

In this section we consider an approach similar to the one adopted for domain influence to a completely different task, where we try to adapt the parser to different sentence constructions, by distinguishing the most usual constituency order, namely Subject–Verb–Object (SVO), and all the others (noSVO). These experiments are taken from [1]. Table 11 reports the dimensions of training and test sets for both patterns. The split of data between the two patterns is strongly unbalanced in favor of the noSVO, corresponding to nearly four times the number of sentences of the other pattern, both in the training and in the test set. This is really a problem, as both training and test will favour the most frequent events.

To overcome such unbalance between the number of sentences in the SVO and in the noSVO data sets, we decided to randomly subsample the noSVO data sets to obtain a training and a test set with exactly the same dimensions of the SVO case. Random sampling is always prone to the risk of unlikely but possible combinations. We therefore repeated this kind of experiments a sufficiently high number of times, namely 20, and averaged the results on the corresponding outputs. We report in Tables 12 and 13 the means of the performance parameters obtained following this approach. Note that significance test is applied to each of the 20 iterations.

In summary, for both constituency and dependency paradigms, we therefore have five models: (i) the “all” model, trained on all training data, (ii) the SVO and (iii) the noSVO models, respectively trained on the SVO and noSVO parts of training data, (iv) the sub-noSVO model, resulting from the average of the 20 models trained on samples extracted from noSVO data, and (v) the “balanced” model whose performance are obtained by averaging the 20 runs of the models trained on the union of the SVO training set and each of the subsampled noSVO training sets.

Table 10 Penn format results

	LR	LP	F_1	EMR
Civil law, baseline				
Berkeley-iteration #5	79.43	79.51	79.47	31.12
Berkeley-iteration #5 $\leq$ 40	83.09	82.94	83.01	37.21
Civil law, specific				
Berkeley-iteration #5	76.00	74.84	75.46	28.87
Berkeley-iteration #5 $\leq$ 40	80.48	79.01	79.74	34.66
Civil law, frequent (>500) *PoS*				
Berkeley-iteration #5	79.40	78.59	78.99	31.16
Berkeley-iteration #5 $\leq$ 40	83.08	82.19	82.63	37.11
Newspaper, baseline				
Berkeley-iteration #5	71.76	71.72	71.74	14.90
Berkeley-iteration #5 $\leq$ 40	75.52	75.33	75.43	18.31
Newspaper, specific				
Berkeley-iteration #5	65.61	63.25	64.41	9.78
Berkeley-iteration #5 $\leq$ 40	70.18	67.46	68.79	11.83
Newspaper, frequent (>500) *PoS*				
Berkeley-iteration #5	72.61	71.65	72.03	14.88
Berkeley-iteration #5 $\leq$ 40	75.89	74.88	75.38	18.26
All together, baseline				
Berkeley-iteration #5	75.49	75.50	75.50	23.01
Berkeley-iteration #5 $\leq$ 40	79.22	79.05	79.14	27.85
All together, specific				
Berkeley-iteration #5	73.42	72.36	72.89	22.01
Berkeley-iteration #5 $\leq$ 40	77.30	75.90	76.59	26.42
All together, frequent (>500) *PoS*				
Berkeley-iteration #5	75.82	75.03	75.43	23
Berkeley-iteration #5 $\leq$ 40	79.41	78.46	78.93	27.80

Table 11 Data sets dimensions

Data set	Pattern	Size
Training set	SVO	646
	noSVO	2,379
	All	3,025
Test set	SVO	110
	noSVO	390
	All	500

Table 12 Constituency parser performance: all comparisons between pairs of models are statistically significant ($p \leq 0.05$)

	All			SVO			noSVO			Sub-noSVO			Balanced		
Penn	LR	LP	F_1	LR	LP	F_1	LR	LP	F_1	LR	LP	F_1	LR	LP	F_1
SVO	81.75	81.37	81.56	72.34	71.49	71.91	79.39	78.10	78.74	69.73	67.95	68.83	76.87	76.41	76.64
noSVO	80.03	80.19	80.11	71.04	70.09	70.56	77.90	77.37	77.64	70.46	69.56	70.01	76.50	76.46	76.48
All	80.51	80.53	80.52	71.42	70.50	70.95	78.32	77.58	77.95	70.26	69.10	69.68	76.60	76.45	76.52
APE															
SVO	77.11	76.96	77.04	69.56	70.21	69.88	78.50	78.90	78.70	67.03	65.36	66.18	74.90	74.03	74.46
noSVO	79.26	79.47	79.36	72.02	72.12	72.07	79.18	78.92	79.05	70.12	69.06	69.59	75.71	75.42	75.57
All	78.69	78.88	78.78	71.57	71.47	71.52	79.02	78.92	78.97	69.34	68.12	68.73	75.51	75.08	75.29

Table 13 Dependency parser performance: labeled accuracy score

	All	SVO	noSVO	Sub-noSVO	Balanced
1-Comp					
SVO	88.44	86.13	83.63	83.49	87.34
noSVO	87.62	86.87	82.43	83.95	85.57
All	87.86	86.65	83.63	83.81	86.08
2-Comp					
SVO	88.84	86.33	86.25	82.62	86.84
noSVO	86.72	86.45	81.11	82.91	84.77
All	87.34	86.41	82.62	82.82	85.38
3-Comp					
SVO	84.60	81.92	86.53	78.09	82.71
noSVO	83.10	82.55	76.87	78.72	80.83
All	83.54	82.86	81.98	78.53	81.38

Statistical significance has been evaluated by using Dan Bikel's Randomized Parsing Evaluation Comparator.[22]

6.1 Constituency Parsing

Parsing performance for both the Penn and the APE formats is depicted in Table 12. The five macro columns correspond to the five different models, obtained by training the parser on: (i) all the training set (All); (ii) only the SVO and (iii) the noSVO parts of the training data (SVO and noSVO) respectively; (iv) by averaging performance on 20 runs made by subsampling the noSVO training set (sub-noSVO); and (v) by considering for training the union of the SVO training set and each of the sets in sub noSVO, and again averaging performance (balanced). For all the models, performance in terms of LP, LR and F_1 is reported. In all the cases, the null hypothesis can be rejected with values of p lower than 0.01 and then the comparisons between performance of all pairs of models result to be statistically significant. Also the standard deviation has been computed for all averaged cases (sub-noSVO and balanced) and its values are always lower than 3. The values have not been reported for providing more compact and readable tables.

First of all, note that the first column represents a sort of baseline, where all available data are exploited. We can see how the addition of more detailed information in the annotation format characterizing APE with respect to Penn does not help parsing, probably because of a data sparsity problem. In fact, we would need a bigger treebank to accurately train the more precise APE labels. In addition to that,

[22] The tool is freely available from http://www.cis.upenn.edu/dbikel/software.html#comparator.

when comparing parsing performance on the SVO and noSVO data sets, we note that the Penn format favors the SVO pattern, while the APE format favors the noSVO pattern. This property is maintained also when training is performed either on SVO or on noSVO data alone, and this is quite surprising, but it probably means that the influence of the annotation is still important. In addition to that, SVO data set contains all and only the sentences containing at least one SVO pattern, and therefore also some noSVO pattern can be included in the SVO data set.

This is no longer the case when we consider the two models obtained by subsampling, namely sub-noSVO and balanced-train. Indeed, the sub-noSVO model always performs better on the corresponding noSVO test set. We can therefore conclude that the better performance of the noSVO model is also related to the fact that the training set is much larger than in the SVO case.

In general, we can conclude that the best choice is to include all the data available in the training set: indeed, this is the case with the best performance on both SVO and noSVO test sets. As a second choice, when the training sets are balanced, the best performance is obtained, as could have been expected, by training the parser on sentences as similar as possible to the ones composing the test set.

6.2 Dependency Parsing

Also for the dependency paradigm, we can note a deterioration in performance when the level of annotation is more fine-grained and, as a consequence, the information to be recognized is more complex. In fact, performance is lower for annotation 3-Comp than for the other two, and for 2-Comp than for 1-Comp. Also, with 3-Comp performance is much less stable than in the other two cases, and this suggests that we are in a data sparsity condition. We therefore focus our analysis on 1-Comp and 2-Comp.

In general, in the dependency case performance remains more or less the same even when training is performed on sentences with a different constituent order with respect to the test set. In fact, when comparing the results obtained with the various settings, no statistically significant variations can be observed.

The fact that performance is only slightly sensitive to the different patterns suggests that the dependency paradigm is more robust than the constituency one with respect to variability in the constituent order and therefore more suitable to MRLs with such feature.

7 Conclusions

The experimental results presented in this chapter aim at demonstrating how data-driven analyzers can be effectively used to assess annotated data sets. We considered the two main paradigms adopted for syntactical analysis, namely constituency

and dependency, but also different annotation designs that make available different amounts of linguistic information. In addition to that, we considered influence to a specific domain and to particular linguistic constructions with a particular focus on issues related to the Italian word order.

When training an analyzer on an annotated data set, a crucial point regards data sparseness, as in general it is very difficult to have a large quantity of accurately annotated data. However, from a different point of view, we could also try to identify an analysis task which can be effectively faced with the available annotated data. In this case, it is important to find a good trade-off between the dimension of the training set and the level of detail of the analysis.

References

1. Alicante, A., Bosco, C., Corazza, A., Lavelli, A.: A treebank-based study on the influence of Italian word order on parsing performance. In: LREC, pp. 1985–1992 (2012)
2. Bosco, C.: A richer annotation schema for an Italian treebank. In: Proceedings of European Summer School on Logic Language and Information, Birmingham, UK (2000), http://www.di.unito.it/~bosco/publicat/esslli00.zip
3. Bosco, C.: Grammatical relation's system in treebank annotation. In: Proceedings of Student Research Workshop of Joint ACL/EACL Meeting, Toulose, France (2001), http://www.di.unito.it/~bosco/publicat/acl-stud-ses-01.zip
4. Bosco, C.: A grammatical relation system for treebank annotation, Ph.D. thesis, University of Torino (2004)
5. Bosco, C.: Multiple-step treebank conversion: from dependency to Penn format. In: Proceedings of Linguistic Annotation Workshop at the ACL'07 (2007)
6. Bosco, C.: Linguistic knowledge extraction from corpus parallel annotations. In: Proceedings of XL Congresso della Società di Linguistica Italiana, Vercelli (2009), http://www.di.unito.it/~bosco/publicat/sli06.zip
7. Bos, J., Bosco, C., Mazzei, A.: Converting a dependency treebank to a categorial grammar treebank for Italian. In: Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories, pp. 27–38. Milan (2009)
8. Bosco, C., Lavelli, A.: Annotation schema oriented evaluation for parsing validation. In: Proceedings of the 9th Workshop on Treebanks and Linguistic Theories (TLT-9), pp. 19–30. Tartu, Estonia (2010)
9. Bosco, C., Mazzei, A., Lavelli, A.: Looking back to the Evalita constituency parsing task: 2007–2011. In: Magnini, B., Cutugno, F., Falcone, M., Pianta, E. (eds.) Evaluation of Natural Language and Speech Tools for Italian—Proceedings of EVALITA 2011, pp. 46–57 (2012)
10. Bosco, C., Lombardo, V.: A relation-schema for treebank annotation. In: A. Cappelli, F.T. (ed.) Advances in Artificial Intelligence, LNCS, vol. 2829. Springer, Berlin (2003), http://www.di.unito.it/~bosco/publicat/aiia-03.zip
11. Bosco, C., Lombardo, V.: Comparing linguistic information in treebank annotations. In: Proceedings of the 5th International Language Resources and Evaluation Conference (2006), http://www.di.unito.it/~bosco/publicat/lrec06.zip
12. Bosco, C., Lombardo, V., Lesmo, L., Vassallo, D.: Building a treebank for Italian: a data-driven annotation schema. In: Proceedings of 2nd International Conference on Language Resources and Evaluation, Athens, Greece (2000), http://www.di.unito.it/~bosco/publicat/lrec00.zip
13. Bosco, C., Mazzei, A., Lombardo, V.: Evalita parsing task: an analysis of the first parsing system contest for Italian. Intell. Artif. **2**(IV), 30–33 (2007)

14. Bosco, C., Mazzei, A., Lombardo, V.: Evalita'09 parsing task: constituency parsers and the Penn format for Italian. In: Proceedings of Evalita'09 (2009)
15. Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A.: Evalita'09 parsing task: comparing dependency parsers and treebanks. In: Proceedings of Evalita'09, Reggio Emilia (2009)
16. Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A., Lesmo, L., Attardi, G., Simi, M., Lavelli, A., Hall, J., Nilsson, J., Nivre, J.: Comparing the influence of different treebank annotations on dependency parsing. In: Proceedings of Language Resources and Evaluation Conference, pp. 1794–1801. Malta (2010)
17. Cheung, J.C., Penn, G.: Topological field parsing of German. In: Proceedings of ACL-IJCNLP'09, pp. 64–72. Singapore (2009)
18. Collins, M., Hajic, J., Ramshaw, L., Tillmann, C.: A statistical parser of Czech. In: Proceedings of the ACL'99 (1999)
19. Corazza, A., Lavelli, A., Satta, G.: An information-theoretic measure to evaluate parsing difficulty across treebanks. ACM Trans. Speech Lang. Process. **9**(4), 7:1–7:31 (2013). http://doi.acm.org/10.1145/2407736.2407737
20. Dell'Orletta, F., Marchi, S., Montemagni, S., Venturi, G.: Domain adaptation for dependency parsing at Evalita 2011. In: Magnini, B., Cutugno, F., Falcone, M., Pianta, E. (eds.) Evaluation of Natural Language and Speech Tools for Italian—Proceedings of EVALITA 2011, pp. 58–69 (2012)
21. Green, S., Manning, C.D.: Better Arabic parsing: Baselines, evaluations, and analysis. In: Proceedings of COLING 2010 (2010)
22. Hajič, J., Böhmová, A., Hajičová, E., Vidová-Hladká, B.: The prague dependency treebank: a three-level annotation scenario. In: Abeillé, A. (ed.) Treebanks: Building and Using Parsed Corpora, pp. 103–127. Kluwer, Amsterdam (2000)
23. Hudson, R.: Word Grammar. Basil Blackwell, Oxford (1984)
24. Jones, B.E.M.: Exploring the role of punctuation in parsing natural text. In: Proceedings of COLING'94, pp. 421–425. Kyoto (1994)
25. Kübler, S., Rehbein, I., van Genabith, J.: TePaCoC a corpus for testing parser performance on complex German grammatical constructions. In: Proceedings of TLT-7, pp. 15–28. Groningen, The Netherlands (2009)
26. Lavelli, A., Hall, J., Nilsson, J., Nivre, J.: MaltParser at the Evalita 2009 dependency parsing task. In: Proceedings of Evalita'09, Reggio Emilia (2009)
27. Lesmo, L.: Use of semantic information in a syntactic dependency parser. In: Magnini, B., Cutugno, F., Falcone, M., Pianta, E. (eds.) Evaluation of Natural Language and Speech Tools for Italian—Proceedings of EVALITA 2011, pp. 13–20 (2012)
28. Lesmo, L.: The rule-based parser of the NLP group of the University of Torino. Intell. Artif. **2**, 46–47 (2007)
29. Lesmo, L.: The Turin University parser at Evalita 2009. In: Proceedings of Evalita'09, Reggio Emilia (2009)
30. Lesmo, L., Lombardo, V., Bosco, C.: Treebank development: the TUT approach. In: Proceedings of ICON02, Mumbai, India (2002), http://www.di.unito.it/~bosco/publicat/icon02lesmo-et-al.zip
31. Nilsson, J., Nivre, J.: MaltEval: An evaluation and visualization tool for dependency parsing. In: Proceedings of LREC'08, pp. 161–166. Marrakech (2008)
32. Nivre, J., Hall, J., Nilsson, J.: MaltParser: A data-driven parser-generator for dependency parsing. In: Proceedings of LREC'06, pp. 2216–2219. Genova (2006)
33. Petrov, S., Klein, D.: Improved inference for unlexicalized parsing. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 404–411. Rochester, New York (April 2007). http://www.aclweb.org/anthology/N/N07/N07-1051
34. Rimell, L., Clark, S., Steedman, M.: Unbounded dependency recovery for parser evaluation. In: Proceedings of Empirical Methods in Natural Language Processing '09, pp. 813–821. Singapore (2009)

Simple Voting Algorithms for Italian Parsing

Alessandro Mazzei

Abstract This paper presents an ensemble system for dependency parsing of Italian: three parsers are separately trained and combined by means of a majority vote. The three parsers are the MATE parser, version 2.0, the DeSR parser, and the MALT parser. We present three experiments showing that a simple voting combination further improves the performances of the parsers.

Keywords Parsing · Ensamble · Combination

1 Introduction

In the last few years Natural Processing Language (NLP) community devoted great attention to the dependency formalisms and many practical NLP systems adopted the dependency parsing [16]. Larger dependency treebanks and more sophisticated parsing algorithms improved the performances of dependency parsers for many languages [13, 21]. For instance, dependency parsing for Italian constantly increased its performances. As reported in the Evalita evaluation campaigns [12], the best scores for Italian dependency parsing (expressed in Labelled Attachment Score, LAS) was 86.94 % in 2007, 88.73 % in 2009, and 91.23 % in 2011 [8]. These results have been obtained by using the Turin University Treebank, a dependency treebank for Italian [7] (see Sect. 3). However, statistical dependency parsing seems to still have room for improving. On the one hand, new promising specific algorithms for learning and classification are emerging; on the other hand, universal machine learning techniques seem to be useful for this specific task. Some algorithms use larger sets of syntactic features (e.g. [10, 19]), while others are trying to apply general techniques *to combine* together the results of various parsers [3, 14, 17, 23, 24, 26]. We designed three experiments on parser combination for Italian that follows both these directions.

A. Mazzei (✉)
Dipartimento di Informatica, Università di Torino,
Corso Svizzera 185, 101049 Torino, Italy
e-mail: mazzei@di.unito.it

R. Basili et al. (eds.), *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*,
Studies in Computational Intelligence 589, DOI 10.1007/978-3-319-14206-7_8

We employed three state of the art statistical parsers, which use sophisticated parsing algorithms and advanced feature sets. The three parsers are the MATE parser[1] [6], the DeSR parser[2] [2], the MALT parser[3] [22]. Moreover, in our system we combined these three parsers by using two very simple voting algorithms [9, 26]. We decided to apply an "off-the-shelf" approach, i.e. we applied each parser with its standard configurations for learning and classification.

Now we give a brief description of the three parsers applied in our experiments, i.e. MATE, DeSR and MALT parsers.

The MATE parser [5, 6] is a development of the algorithms described in [10, 15]. It basically adopts the second order maximum spanning tree dependency parsing algorithm. In particular, Bohnet exploits *hash kernel*, a new parallel parsing and feature extraction algorithm that improves the accuracy as well as the parsing speed [6]. The MATE performances on English and German, which are 90.14 and 87.64 % respectively (LAS), posed this parser at the state of the art for these languages [1, 6, 13].

The DeSR parser [2] is a transition (shift-reduce) dependency parser similar to [25]. It builds dependency structures by scanning input sentences in left-to-right and/or right-to-left direction. For each step, the parser learns from the annotated dependencies if to perform a shift or to create a dependency between two adjacent tokens. DeSR can use different set of rules and includes additional rules to handle non-projective dependencies. The parser can choose among several learning algorithms (e.g. Multi Layer Perceptron, Simple Vector Machine), providing user-defined feature models. In our experiments we adopted for DeSR the Multi Layer Perceptron algorithm, which is the same configuration that the parser exploited when it won the Evalita 2009 competition.

The MALT parser [22] implements the transition-based approach to dependency parsing too. In particular MALT has two components: (1) a (non-deterministic) transition system that maps sentences to dependency trees; (2) a classifier that predicts the next transition for every possible system configuration. MALT performs a greedy deterministic search into the transition system guided by the classifier. In this way, it is possible to perform parsing in linear time for projective dependency trees and quadratic time for arbitrary (non-projective) trees [20]. MALT has several built-in transition systems, but in our experiments we adopted just the standard "Nivre arc-eager" system, that builds structure incrementally from left to right. Moreover, we use the standard classifier provided by MALT, i.e. the SVM (Simple Vector Machine) basic classifier on the standard "NivreEager" feature model.

To our knowledge this is the first work that experimented the MATE parser on Italian, while DeSR and MALT parsers have been used in many occasions on Italian (e.g. [4, 17]), reaching the best results in several contests. In the next sections we describe our approach for ensemble parsing (Sect. 2) and we report the results of three experiments (Sect. 3), before concluding the paper (Sect. 4).

[1] http://code.google.com/p/mate-tools/

[2] http://sites.google.com/site/desrparser/

[3] http://maltparser.org/

2 The Combination Algorithms

In order to combine the three parsers we used two very simple algorithms, COM1 and COM2 (see Algorithms 1 and 2), both implemented in the PERL programming language. These algorithms have been previously experimented in [24, 26]. The main idea of the COM1 algorithm is to do a democratic voting among the parsers. For each word[4] of the sentence, the dependency (the parent and the edge label) assigned to the word by each parser is compared: if at least two parsers assign the same dependency, the COM1 algorithm selects that dependency. In the case that each parser assigns a different dependency to the word, the algorithm selects the dependency assigned by the "best parser". As noted by [26], who use the name *voting* for COM1, this is the most logical decision if it is possible to identify a priori the "best parser", in contrast to the more democratic random choice.

```
foreach sentence do
    foreach word in the sentence do
        if DependencyParser2(word) == DependencyParser3(word) then
            DependencyParser-COM1(word) := DependencyParser2(word)
        else
            DependencyParser-COM1(word) := DependencyParser1(word)
        end
    end
end
```

Algorithm 1: The combination algorithm COM1, that corresponds to the *voting* algorithm reported in [26]

The COM2 algorithm is a variation of the COM1. COM1 is a single word combination algorithm that does not consider the whole dependency structure. This means that incorrect dependency trees can be produced by the COM1 algorithm: cycles and multiple roots can *corrupt* the "treeness" of the structure. The solution that we adopt in the COM2 algorithm is naive: if the tree produced by the COM1 algorithm for a sentence is corrupted, then the COM2 returns the tree produced by the "best parser". Again, similarly to [26], who use the name *switching* for COM2, this is the most logical decision when there is an emerging best parser from a development data set.

3 Experimental Results

We applied our approach for parsing combination in three experiments. In the first experiment we use the datasets provided in the SPLeT competition [11], in the second experiment we used the datasets provided in the Evalita 2011 competition [8], and in the third experiment we used both the datasets. For all the experiments we used two machines. A powerful Linux workstation, equipped with 16 cores, processors 2 GHz, and 128 GB ram has been used for the training of the MATE parser, that is the

[4] In this paper we use the term *word* in a general sense, as synonym of *token*.

most computationally expensive system: on this machine the average training time for MATE was 8 h.

```
foreach sentence do
    foreach word in the sentence do
        if DependencyParser2(word) == DependencyParser3(word) then
            DependencyParser-COM2(word) := DependencyParser2(word)
        else
            DependencyParser-COM2(word) := DependencyParser1(word)
        end
    end
    if TREE-COM2(sentence) is corrupted then
        TREE-COM2(sentence) := TREE-PARSER1(sentence)
    end
end
```

Algorithm 2: The combination algorithm COM2, that corresponds to the *switching* algorithm reported in [26]

Another Linux workstation equipped with a single processor 1 GHz, and 2 GB ram has been used for the training of the DeSR and MALT parsers: that usually required a couple of hours. This machine has been used for testing all the systems: this phase required several minutes for MATE parser and few minutes for MALT and DeSR parsers. MALT and DeSR parsers accept as input the CONLL-07 format, that is the format provided by the SPLeT organizers. In contrast, MATE accepts the CONLL-09 format: simple conversions scripts have been implemented to manage this difference.

3.1 The SPLeT Experiment

In the SPLeT experiment, a first run was performed in order to evaluate the "best parser" in the COM1 and COM2 algorithms. We used the ISST training[5] (71,568 words, 3,275 sentences) as training set and the ISST development[6] (5,165 words, 231 sentences) as development set. The first row in Table 1 shows the results of the three parsers in this first experiment. MATE parser outperforms the DeSR and MALT parsers: MATE does ∼3 % better than DeSR and ∼5 % better than MALT. On the basis of this result, we used MATE as our "best parser" in the combination algorithms (cf. Sect. 2). COM1 and COM2 reach the score of 82.54 and 82.36 % respectively, and so both combination algorithms improve the performances of the MATE parser close to the 0.5 %.

[5] `File: it_isst_train.splet`.

[6] `File: it_isst_test.splet`.

Table 1 The performances (LAS score) of the three parsers, their simple combination (COM1 and COM2), their blended combination (Blended$_{W_2}$, Blended$_{W_3}$, Blended$_{W_4}$) on the SPLeT test set, development set, Regional laws set

	MATE	DeSR	MALT	COM1	COM2	BL$_{W_2}$	BL$_{W_3}$	BL$_{W_4}$
DevSet	81.92	78.99	77.04	82.54	82.36	81.45	82.54	82.63
TestSet	82.57	78.68	77.98	83.20	**83.08**	82.23	83.15	83.24
NatReg	75.76	70.66	70.33	76.28	75.88	74.78	76.07	75.97

In a second run, we used the whole ISST as training set[7] (total 76,733 words, 3,506 sentences) and we used the blind file provided by the organizers as test set[8] (5,662 words, 240 sentences, European Directives Laws). The second row in Table 1 shows the results of the three parsers in this second experiment: the value 83.08 %, produced by the COM2 algorithm, is the final result of our participation to the SPLeT shared task [18]. Note that there is a ∼0.1 % difference between the COM1 and COM2 results: similar to [24, 26] we have 10 corrupted trees in the test set, i.e. ∼4 % of the total (240 sentences). In Table 2 we detailed the results of the three parsers in the SPLeT experiment on the basis of their agreement. When the three parsers agree on the same dependency (Table 2, first row), this happens on ∼72 % of the words, they have a very high LAS score, i.e. 95.6 %. Moreover, DeSR and MALT parsers do better than the MATE parser only when they agree on the same dependency (Table 2, second row). The inspection of the other rows in Table 2 shows that COM1 algorithms has the best possible performance w.r.t. the voting strategy. In other words, COM1 selects all the parser combinations that correspond to higher value of LAS score (cf. the discussion on *minority dependencies* in [24]).

In a third run, we again use the whole ISST as training set[9] (total 76,733 words, 3,506 sentences), but we use the NatReg file provided by the organizers as test set[10] (5,194 words, 119 sentences, Regional Laws of Piedmont Region). The third row in Table 1 shows the results of the three parsers in this third run: in this case we have 75.88 % for COM2 algorithm. This lower result can be advocated to the different nature of the domain. It is interesting to note that in this experiment MALT and DeSR parsers give similar results (∼70 %), while the MATE parser still outperforms them by ∼5 %.

[7] `File:` *it_isst_train.splet* and *it_isst_test.splet*.

[8] `File:` *it_EULaw_test_blind.splet*.

[9] `File:` *it_isst_train.splet* and *it_isst_test*. *splet*.

[10] `File:` *it_NatRegLaw_test_blind.splet*.

Table 2 The detailed performances (LAS score) of the three parsers and their simple combination on the SPLeT blind set, corresponding to the first row of the Table 1

Scores	Frequency
MATE == DeSR == MALT **95.6**	**71.99**
MATE != **DeSR** == **MALT** 30.7 **45.8**	**4.20**
MATE == **DeSR** != MALT **67.2** 14.4	**7.70**
MATE == **MALT** != DeSR **59.1** 20.0	**8.21**
MATE != DeSR != MALT **31.1** 14.5 16.3	**7.89**

3.2 The EVALITA Experiment

We performed a second experiment on two different training and test sets belonging to a different Italian Treebank, which has a different PoS tag set and a different dependency label set. We used for learning the Evalita 2011 Development Set[11] (93,987 words, 3,452 sentences; balanced corpus of newspapers, laws, wikipedia) and we use for testing the Evalita 2011 test set[12] (7,836 words, 300 sentences; balanced corpus): these sets have been annotated according to the format of the Turin University Treebank [8]. The first row in Table 3 shows the results of the three parsers in this experiment: in this case we have 89.16 % for COM2. It is interesting to note that the improvement of the COM2 algorithm with respect to the MATE parser is only ∼0.1 %. In Table 4 we detailed the results of the three parsers in this run on the basis of their agreement. Again, when the three parsers agree on the same dependency (Table 4, first row) which happens for ∼78 % of the words, they have a very high LAS score, i.e. 96.6 %. In contrast with the SPLeT experiment, here we do not have a rel-

Table 3 The performances (LAS score) of the MATE, DeSR, MALT, Parsit, UniPi, FBKirst and UniTo parsers, their simple combination (COM1 and COM2), their blended combination ($BLended_{W_2}$, $BLended_{W_3}$, $BLended_{W_4}$) on the Evalita 2011 test

MATE	DeSR	MALT	COM1	COM2	BL_{W_2}	BL_{W_3}	BL_{W_4}
89.07	86.26	80.76	89.19	89.16	88.03	89.19	89.19
Parsit	UniPi	FBKirst	COM1	COM2	BL_{W_2}	BL_{W_3}	BL_{W_4}
91.23	89.88	88.62	91.95	92.04	91.12	91.97	91.93
Parsit	UniPi	UniTo	COM1	COM2	BL_{W_2}	BL_{W_3}	BL_{W_4}
91.23	89.88	85.34	92.54	92.50	91.39	92.57	92.65

[11] `File:` *evalita201_ train.conll*.

[12] `File:` *evalita2011_ test.conll*.

Table 4 The detailed performances (LAS score) of the MATE, DeSR and MALT parsers and their combination on the Evalita 2011 test set

Scores	Frequency
MATE == DeSR == MALT **96.6**	**78.39**
MATE != **DeSR** == **MALT** 35.2 **38.8**	**3.38**
MATE == **DeSR** != MALT **82.0** 7.2	**9.17**
MATE == **MALT** != DeSR **63.3** 19.6	**4.27**
MATE != DeSR != MALT **40.7** 18.4 7.9	**4.78**

evant improvement when DeSR and MALT parsers do better than the MATE parser, i.e. only when they agree on the same dependency (Table 4, second row). In other words, on the SPLeT test set, the COM1[13] algorithm does much better than MATE since DeSR and MALT parsers have a good performance (45.8 vs. 30.7 %) when they do not agree with the MATE parser: this is not true for the EVALITA experiment, where DeSR and MALT have 38.8 % while MATE has 35.2 %.

In order to evaluate the COM1 and COM2 algorithms in a more general context, we performed two new runs on the EVALITA dataset by using other parsers. We used four parsers that have participated to the Evalita 2011 competition [8]. In the first run (second row in Tables 3 and 5) we combined the Parsit,[14] the UniPi[15] and the

Table 5 The detailed performances (LAS score) of the Parsit, UniPi and FBKirst parsers on the Evalita 2011 test set

Scores	Frequency
Parsit == UniPi == FBKirst **97.7**	**85.15**
Parsit != **UniPi** == **FBKirst** 37.7 **49.0**	**6.34**
Parsit == **UniPi** != FBKirst **75.9** 9.4	**3.59**
Parsit == **UniPi** != FBKirst **66.8** 19.5	**2.57**
Parsit != UniPi != FBKirst **52.3** 16.1 12.6	**7.89**

[13] The same consideration hold for COM2: in the second experiment there are just 8 corrupted trees.

[14] http://www.parsit.it.

[15] The UniPi parser is the DeSR parser tuned for this specific competition.

Table 6 The detailed performances (LAS score) of the Parsit, UniPi and UniTo parsers on the Evalita 2011 test set

Scores	Frequency
Parsit == UniPi == UniTo **98.2**	**80.92**
Parsit != **UniPi** == **UniTo** 29.6 **58.3**	**4.57**
Parsit == **UniPi** != UniTo **81.7** 9.3	**7.82**
Parsit == **UniPi** != UniTo **72.1** 15.5	**2.96**
Parsit != UniPi != UniTo **49.6** 23.2 8.3	**3.73**

FBKirst[16] parsers, i.e. the best scored systems in the competition. In the second run (third row in Tables 3 and 6) we combined the Parsit, UniPi and the UniTo parsers, i.e. two statistical parsers and one rule-based parser. From Table 3 we can note that the best result (92.54 %) is obtained by the COM1 in the second run, i.e. when the UniTo parser belongs to the ensemble. Comparing the second rows in Table 5 and in the Table 6 we can explain this result. There is a relevant improvement when UniPi and UniTo parsers do better than the Parsit parser, i.e. the COM1 algorithm do much better than Parsit since UniPi and UniTo parsers have a good performance (29.6 versus 58.3 %) when they do not agree with the Parsit parser. This result confirms that the performance of the parsing combination depends on the "diversity" of the parsers involved rather than on the absolute score of each single parser.

3.3 Parsing Combination Versus Re-parsing Experiment

Similar to [26], we designed the COM2 algorithm since COM1 can produce corrupted dependency trees. COM2 tests the correctness of the tree and in the case of corruption returns the dependency structure produced by the "best parser" of the ensemble. We hypothesized that this strategy can produce good results in our system since one of the parser of the ensemble drastically outperforms the others. However, some more general solution to the tree-corruption problem have been proposed: the re–parsing strategy [3, 14, 23]. In re–parsing, a new, not corrupted, dependency tree is produced by taking into account the trees produced by each parser of the ensemble. Attardi and Dell'Orletta proposed an approximate top-down algorithm that starts by selecting the highest-scoring root node, then the highest-scoring children and so on [3]. Sagae and Lavie together with Hall et al. proposed a two-steps algorithm: (1) to create a graph

[16] The FBKirst parser is an ensemble combination of the MALT parser.

by merging all the structures produced by the parser on the ensemble, and (2) to extract the most probable dependency spanning tree from this graph [14, 23].

Surdeanu and Manning provided experimental evidence that re–parsing algorithms are a good choice for practical ensemble parsing in out domains [24]: in order to confirm this hypothesis we performed a third experiment on both the SPLeT and EVALITA datasets by using the "MaltBlender" tool [14]. In Tables 1 and 3 the columns $\mathbf{BL}_{W_2}$, $\mathbf{BL}_{W_3}$, $\mathbf{BL}_{W_4}$ report the application of the algorithm described in [14]. There are three weighting strategies: the results of the three parsers are equally weighted (W_2); the three parsers are weighted according to the total labeled accuracy on a held-out development set (W_3); the parsers are weighted according to labeled accuracy per coarse grained PoS tag on a held-out development set (W_4). For the first, the second and the third runs of the SPLeT experiment (Table 1), the held-out development set is the SPLeT development set; for the EVALITA experiment (Table 3), the held-out development set is the Evalita 2011 test set.

Three evidences seems to emerge from the third experiment: (1) the re–parsing strategy always performs slightly better than COM2 algorithm but not always better than COM1 algorithm; (2) there is no winning weighting strategy for re-parsing; (3) it does not seem that blending performs better out domain than in domain.

4 Conclusions

In this paper we described three parsing experiments on three parsers, i.e. the MATE, the DeSR and the MALT parsers. The first emerging issue by these experiments is that the MATE parser has a very good performance on Italian ISST treebank, both in domain and out domain, reaching very good scores. The EVALITA experiment confirms that similar results can be obtained on the Turin University Treebank. The second emerging issue is that very simple combination algorithms, as well as more complex blending algorithms, can furthermore improve performance also in situations where one parser outperforms the others.

In future research we plan to repeat our experiments on a larger set of parsers. In particular, on the basis of the results emerged by the EVALITA experiment, i.e. that "diversity" is an important value in combining parsers, we want to perform more tests on the combination of statistical parsers with rule based parsers.

References

1. Anders, B., Bernd, B., Hafdell, L., Nugues, P.: A high-performance syntactic and semantic dependency parser. In: Coling 2010: Demonstrations, pp. 33–36. Coling 2010 Organizing Committee, Beijing, China (August 2010), http://www.aclweb.org/anthology/C10-3009
2. Attardi, G.: Experiments with a multilanguage non-projective dependency parser. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X),

pp. 166–170. Association for Computational Linguistics, New York (June 2006), http://www.aclweb.org/anthology/W/W06/W06-2922
3. Attardi, G., dell'Orletta, F.: Reverse revision and linear tree combination for dependency parsing. In: HLT-NAACL, pp. 261–264 (2009)
4. Attardi, G., Simi, M., Zanelli, A.: Tuning DeSR for the Evalita 2011 Dependency Parsing. In: Working Notes of EVALITA 2011. CELCT a r.l. (2012) ISSN 2240–5186
5. Bohnet, B.: Efficient parsing of syntactic and semantic dependency structures. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL'09), Shared Task, pp. 67–72. Association for Computational Linguistics, Stroudsburg (2009), http://dl.acm.org/citation.cfm?id=1596409.1596421
6. Bohnet, B.: Top accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 89–97. Coling 2010 Organizing Committee, Beijing, China (August 2010), http://www.aclweb.org/anthology/C10-1011
7. Bosco, C., Lombardo, V.: Dependency and relational structure in treebank annotation. In: Proceedings of the COLING'04 workshop on Recent Advances in Dependency Grammar. Geneve, Switzerland (2004), http://www.di.unito.it/~bosco/publicat/dependency-coling04.zip
8. Bosco, C., Mazzei, A.: The Evalita 2011 parsing task: the dependency track. In: Working Notes of EVALITA 2011. CELCT a r.l. (2012) ISSN 2240–5186
9. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
10. Carreras, X.: Experiments with a higher-order projective dependency parser. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 200, pp. 957–961 (2007), http://www.aclweb.org/anthology/D/D07/D07-1101
11. Dell'Orletta, F., Marchi, S., Montemagni, S., Plank, B., Venturi, G.: The SPLeT-2012 Shared Task on Dependency Parsing of Legal Texts. In: SPLeT 2012—Fourth Workshop on Semantic Processing of Legal Texts (SPLeT 2012)—First Shared Task on Dependency Parsing of Legal Texts (2012)
12. EVALITA 2011 Organization Committee: Working Notes of EVALITA 2011. CELCT a r.l (2012)
13. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The conll-2009 shared task: syntactic and semantic dependencies in multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning CoNLL'09, Shared Task, pp. 1–18. Association for Computational Linguistics, Stroudsburg (2009), http://dl.acm.org/citation.cfm?id=1596409.1596411
14. Hall, J., Nilsson, J., Nivre, J., Eryigit, G., Megyesi, B., Nilsson, M., Saers, M.: Single malt or blended? A study in multilingual parser optimization. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 933–939 (2007), http://www.aclweb.org/anthology/D/D07/D07-1097
15. Johansson, R., Nugues, P.: Dependency-based syntactic-semantic analysis with propbank and nombank. In: Proceedings of the Twelfth Conference on Computational Natural Language Learning CoNLL'08, pp. 183–187. Association for Computational Linguistics, Stroudsburg (2008), http://dl.acm.org/citation.cfm?id=1596324.1596355
16. Kübler, S., McDonald, R.T., Nivre, J.: Dependency Parsing. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Rafael (2009)
17. Lavelli, A.: An ensemble model for the EVALITA 2011 dependency parsing task. In: Working Notes of EVALITA 2011. CELCT a r.l. (2012). ISSN 2240–5186
18. Mazzei, A., Bosco, C.: Simple parser combination. In: SPLeT 2012—4th Workshop on Semantic Processing of Legal Texts (SPLeT 2012)—First Shared Task on Dependency Parsing of Legal Texts, pp. 57–61 (2012)
19. McDonald, R., Pereira, F.: Online learning of approximate dependency parsing algorithms. In: Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006), vol. 6, pp. 81–88 (2006)

20. Nivre, J.: Algorithms for deterministic incremental dependency parsing. Comput. Linguist. **34**(4), 513–553 (2008)
21. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932 (2007), http://www.aclweb.org/anthology/D/D07/D07-1096
22. Nivre, J., Hall, J., Nilsson, J.: Maltparser: a data-driven parser-generator for dependency parsing. In: Proceedings of LREC-2006, vol. 6, pp. 2216–2219 (2006)
23. Sagae, K., Lavie, A.: Parser combination by reparsing. In: Moore, R.C., Bilmes, J.A., Chu-Carroll, J., Sanderson, M. (eds.) HLT-NAACL. The Association for Computational Linguistics (2006)
24. Surdeanu, M., Manning, D.C.: Ensemble models for dependency parsing: cheap and good? In: NAACL. The Association for Computational Linguistics (2010)
25. Yamada, H., Matsumoto, Y.: Statistical dependency analysis with support vector machines. In: Proceedings of IWPT, vol. 3 (2003)
26. Zeman, D., Žabokrtskỳ, Z.: Improving parsing accuracy by combining diverse dependency parsers. In: International Workshop on Parsing Technologies, pp. 171–178. Association for Computational Linguistics, Vancouver (2005)