Discriminative Reranking for Spoken Language Understanding

Marco Dinarelli, Alessandro Moschitti, Giuseppe Riccardi

Discriminative Reranking for Spoken Language Understanding

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi, *Fellow, IEEE*

***Abstract*—Spoken Language Understanding (SLU) is concerned with the extraction of meaning structures from spoken utterances. Recent computational approaches to SLU, e.g. Conditional Random Fields (CRF), optimize local models by encoding several features, mainly based on simple n-grams. In contrast, recent works have shown that the accuracy of CRF can be significantly improved by modeling long-distance dependency features. In this paper, we propose novel approaches to encode all possible dependencies between features and most importantly among parts of the meaning structure, e.g. concepts and their combination. We rerank hypotheses generated by local models, e.g. Stochastic Finite State Transducers (SFSTs) or Conditional Random Fields (CRF), with a global model. The latter encodes a very large number of dependencies (in the form of trees or sequences) by applying kernel methods to the space of all meaning (sub) structures. We performed comparative experiments between SFST, CRF, Support Vector Machines (SVMs) and our proposed discriminative reranking models (DRMs) on representative conversational speech corpora in three different languages: the ATIS (English), the MEDIA (French) and the LUNA (Italian) corpora. These corpora have been collected within three different domain applications of increasing complexity: informational, transactional and problem-solving tasks, respectively. The results show that our DRMs consistently outperform the state-of-the-art models based on CRF.**

I. Introduction

SPOKEN Language Understanding is concerned with the task of mapping utterances into meaning representations based on semantic constituents. These are instantiated by word sequences and are often referred to as concepts, attributes or semantic tags. Traditionally grammar-based methods have been used but more recently machine learning approaches to semantic structure computation have received a lot of attention, due to their performance and incremental learning ability [1]. State-of-the-art learning algorithms, e.g. CRF [2], are successfully applied to perform conceptual tagging at word level; these models exploit mainly features based on n-grams.

One drawback of the above-mentioned methods is that the word dependencies captured by such features have their scope constrained by the locality of the target word. To overcome this limitation, CRF models capable of capturing long-dependency features, i.e. the arbitrary interactions and inter-dependencies that exist in the observation sequences, have been applied, e.g. [3]–[6]. The number of all such possible features is extremely large, thus the subset of relevant features must be specified and designed in advance, e.g. according to a feature-generating scheme based on domain knowledge.

M. Dinarelli worked on this research during his PhD Student at the Department of Information Engineering and Computer Science.

A. Moschitti and G. Riccardi are with the Department of Information Engineering and Computer Science, University of Trento, Italy

In this paper, we contribute on the above-mentioned research in different ways: first, we effectively model dependencies between features and most importantly among parts of the meaning structure, e.g. concepts, features and their combinations. To extract the dependencies from the meaning structure, this must be available at learning time. Thus we approach SLU by reranking the hypotheses generated by a baseline model: in our case we use two different local models, i.e. SFSTs [7] and CRF [2]. Our discriminative reranking is modeled with SVMs, which also enable the use of kernel-based learning [8].

Second, we exploit kernel methods (e.g. see [9]) to generate the space of all possible dependencies between features and concepts at any distance in the observation. More specifically, we design sequential and tree structures to describe the conceptual meaning structure and compactly represent semantic and syntactic dependencies [10]–[13]. Then, we apply tree and sequence kernels developed in [14]–[20] to blow up the above-mentioned structures in the space of substructures. These correspond to dependency features between any arbitrary number of basic features and concepts at any distance.

Third, since rerankers may be limited by the quality of the small number (e.g. generally in the order of ten) of hypotheses produced by the local model, we propose a semantic inconsistency metric (SIM) capable of selecting accurate hypotheses from an initial large set. Although such metrics is domain specific, it can be easily adapted to other natural language processing tasks.

Finally, we improve our DRMs by designing a simple but effective meta-model selection strategy. For each utterance, the strategy chooses to apply or not reranking by comparing the classification confidence of the local and reranker models.

Regarding the empirical validation, we tested our DRMs on different domains, languages and noisy conditions. More precisely, we used two different kinds of input: manual transcriptions of spoken sentences and automatic transcriptions generated by Automatic Speech Recognition systems (ASR). We combine them with three of the most relevant SLU annotated corpora in different languages: the well-known ATIS corpus [21], the French MEDIA corpus [22] and the Italian conversational corpus acquired within the European project LUNA [10].

Such corpora are very different with respect to the task they address (informational, transactional and problem-solving tasks), speaking styles and the semantic complexity in terms of number of semantic classes and characteristics of user utterances. Therefore, they help us to consider SLU in several conditions: domain, language and style of spoken utterances.

	train.		test	
# turns	4,978		893	
	words	conc.	words	conc.
# tok.	52,178	16,547	8,333	2,800
# voc.	1,045	80	484	69
# OOV%	–	–	1.0	0.1

TABLE I
STATISTICS OF THE ATIS TRAINING AND TEST SETS USED IN THE EXPERIMENTS

	train.		dev.		test	
# sent.	12,908		1,259		3,005	
	words	concepts	words	concepts	words	concepts
# tok.	94,466	43,078	10,849	4,705	25,606	11,383
# voc.	2,210	99	838	66	1,276	78
# OOV%	–	–	1.33	0.02	1.39	0.04

TABLE II
STATISTICS OF THE MEDIA TRAINING, DEVELOPMENT AND EVALUATION SETS USED FOR ALL EXPERIMENTS

The results show that our best DRMs significantly improve the state-of-the-art models based on CRF, across all domains/experiments, e.g. up to 2 and 3 absolute percent points (about 7% and 10% relative error reduction) on MEDIA and LUNA, respectively. The less accurate FST model is improved by 6 points (about 10% relative error reduction).

The paper is organized as follows. In Section II, we define the problem of SLU in the context of spoken conversational systems by also illustrating the corpora studied in this paper. Section III illustrates our proposed DRMs whereas Section IV shows their evaluation across different domains, corpora and languages. Finally, Section V provides the final remarks in the light of previous work.

II. SPOKEN LANGUAGE UNDERSTANDING IN ATIS, MEDIA AND LUNA CORPORA

The novelty of our work relies on the design of new reranking models, which learn to sort the annotation hypotheses generated by SLU baseline models. The SLU hypotheses refer to a meaning representation of spoken utterances and they include a complete mapping from words into semantic categories[1] (or concepts). This process is typically divided in two steps: text segmentation and labeling. The concept lexicon for the latter is acquired from a knowledge base such as a relational database or domain ontology of a target application task. In the case of the ATIS corpus [21], the knowledge base is a relational database while for the MEDIA [22] and the LUNA corpora [10] a domain ontology was designed (see [23] for the LUNA ontology). In the following, we describe the typical format of annotation hypotheses for the corpora above along with the description of the segmentation and labeling phases, where the latter also includes the extraction of attribute-values.

A. Description of the SLU Corpora

The Air Travel Information System (ATIS) corpus [21] has been used for the last decade to evaluate models of Automatic Speech Recognition and Understanding. It includes speech uterarnces acquired via a Wizard-of-Oz (WOZ) approach, where users ask for flight information. Statistics for this corpus, i.e. turns, tokens (tok.) constituted by words or concepts (conc.), vocabulary items (voc.), percentage of out of vocabulary token (OOV%) for training (train.) and test sets, are reported in Table I.

[1]Their relations are useful to form an interpretation exploitable in a conversation context [1]

	train.		dev.		test	
# sent.	3,171		387		634	
	words	concepts	words	conc.	words	concepts
# tok.	30,470	18,408	3,764	2,258	6,436	3,783
# voc.	2,386	42	777	38	1,059	38
# OOV%	–	–	422	0.0	3.68	0.0

TABLE III
STATISTICS OF THE LATEST VERSION OF THE LUNA TRAINING, DEVELOPMENT AND EVALUATION SETS USED FOR ALL EXPERIMENTS.

The corpus MEDIA has been collected within the French project MEDIA-EVALDA [22] for development and evaluation of spoken understanding models and linguistic studies. The corpus is composed of 1,257 dialogs (from 250 different speakers) acquired with a WOZ approach in the context of hotel room reservations and tourist information. Statistics on transcribed and conceptually annotated data are reported in Table II. In this case, the corpus is divided in sentences (sent.).

The LUNA corpus, produced in the homonymous European project, is the Italian corpus of conversational speech. It has been collected in a contact center providing help-desk support for software and hardware [10]. The data is organized in transcriptions and annotations of speech based on a new multi-level protocol. Here, we provide for the first time results on the latest version of the corpus. The data used for our experiments is extracted from 723 Human-Machine dialogs (HM) acquired with a WOZ approach. The data has been split, with respect to sentences, in training, development and test sets. Statistics of this corpus are reported in Table III.

1) Examples of SLU for different corpora: The following sections show the conceptual annotation available for the three mentioned corpora, where the difference and complexity are highlighted.

a) ATIS: Given the following sentence *"I would like a flight from Phoenix to San Diego on April First"*, an example of the concept annotation of the ATIS corpus is:

null{I would like a flight from} **departure_city**{Phoenix} **null**{to} **arrival_city**{San Diego} **null**{on} **departure_date.month**{April} **departure_date.day_number**{first}

where **departure_city**, **arrival_city** are domain concepts used for departure and arrival cities, respectively. **departure_date.month** and **departure_date.day_number** are used for departure date month and day, respectively. **null** is the concept tag mapping words not covered by the knowledge base.

b) MEDIA: As an example taken from the MEDIA corpus, let us consider the sentence: *"Je veux une chambre double"* that translates to *"I want a double room"*, a semantic representation is:

null{Je veux} **nb_chambre**{une} **chambre_type**{chambre double}

where **nb_chambre** and **chambre_type** are domain concepts modeling number and type of rooms, respectively.

c) LUNA: Given the transcription: *"Buongiorno io ho un problema col mio monitor da questa mattina non riesco piu' ad accenderlo"* from the LUNA corpus (*"Good morning*

I have a problem with my screen, I cannot turn it on any more since this morning"), an example of the corresponding semantic annotation is:

null{Buongiorno io ho} **HardwareProblem.type**{un problema}
Peripheral.type{col mio monitor} **Time.relative**{da questa mattina}
HardwareOperation.negate{non riesco} **null**{piu'}
HardwareOperation.operationType{ad accenderlo}

In this case, the domain concepts are **HardwareProblem.type**, **Peripheral.type**, used to model types of hardware problem and of peripheral devices, **Time.relative**, used for relative time expressions (this morning, this afternoon, two days ago etc.), **HardwareOperation.negate** and **HardwareOperation.operationType**, used to describe actions performed on hardware components.

Note that in the Italian corpus concepts are expressed as fields of a class, so that different concepts belonging to the same class can be merged to construct more general and abstract semantic objects like **Hardware**. As shown in [23], this representation can be exploited to perform semantic analysis based on domain ontology relations.

2) Differences among corpora: Hereafter, we report shared and different corpus characteristics:

First, application domain. From this point of view ATIS and MEDIA are rather similar, the former is a corpus of flight information and reservation whereas the latter is a corpus of hotel information and reservation.

Second, data collection paradigm. All corpora have been acquired with a WOZ approach but with a different setup. In ATIS the data acquisition unit is a single turn, where the users ask flight information, whereas in MEDIA and LUNA the units are entire dialogs.

Third, size of the data. LUNA is the smallest corpus ($3,171$ turns for training), while MEDIA is (almost thirteen thousand sentences for training). ATIS is in the middle with roughly five thousand sentences for training.

Finally, task complexity. It is usually measured in terms of number of concepts with respect to the size of the available training data. From this point of view LUNA with only 42 concepts is the simplest task. ATIS and MEDIA have a comparable complexity since the former includes 69 concepts whereas the latter contains 65 concepts. Nevertheless MEDIA is much more complex since some of its concepts have different specifiers and modes (see [22]). Thus the real number of tags to be recognized in MEDIA increases to 99.

Moreover, it should be noted that the automatic annotation of ATIS can be easier than in other corpora for two reasons: (i) most sentences have the form: *"Information Request about"* flights from DEPARTURE_CITY to ARRIVAL_CITY TIME, where *"Information Request about"* is one of the several ways of asking information, *DEPARTURE_CITY* and *ARRIVAL_CITY* are the names of two cities and *TIME* is the specification of a day and/or hour of departure. This kind of sentences with small variations constitute more than 90% of the corpus. (ii) In the data available for the SLU task on ATIS, which is the same used in [24] and in [25], concepts are always associated with a single token[2] so there is no need of segmenting them using BIO-like markers (as shown in Section II). For example, the previous ATIS sentence, using the annotation style of the Media or Italian LUNA corpora, would be annotated as:

null{I would like a flight} **departure_city**{from Phoenix} **arrival_city** {to San Diego} **departure_date.month**{April}
departure_date.day_number{first}

That is, the concepts **departure_city** and **arrival_city** would have a span of two and three words respectively. In contrast, ATIS only concerns with the problem of token labeling, there is no need to carry out concept segmentation. For these reasons, our work on ATIS only relates to concept labeling: the segmentation can be attained with the deterministic processing of matching word surface forms.

The task complexity is also affected by the characteristics of utterances. ATIS and MEDIA were acquired with a WOZ approach with optimal environmental setup (high quality microphones and absence of noise in the channel) whereas LUNA was acquired from customers calling call center operators. Additionally, (i) utterances in the LUNA corpus are spontaneous, thus including typical phenomena such as disfluencies, ill-formed transcriptions and noisy signals; (ii) the annotation of the turns in the Italian LUNA corpus was done taking into account turn context. The same words can be annotated with a different concept in case the context is different. For example, the phrase "it is not working" can be a "**HardwareOperation**" in case it refers to a "**Peripheral**", while it is a "**SoftwareOperation**" if it refers to "**Software**". For these characteristics, even if the number of concepts to be recognized is smaller, the LUNA corpus is not simpler than the other two.

B. Concept Segmentation and Labeling

1) Concept Segmentation: One important phase in the SLU process is the concept chunking, i.e. concepts can span over more than one word. In order to have a one-to-one association between words and concepts, the beginning of a concept is distinguished from its other components using markers equivalent to those of the *BIO* notation [26]. In particular the *Outside* marker (*O*) is replaced by the **null** tag introduced before. Using this notation the semantic representation for the example shown above would be:

null{Buongiorno io ho} **HardwareProblem.type-B**{un}
HardwareProblem.type-I{problema} **Peripheral.type-B**{col}
Peripheral.type-I{mio} **Peripheral.type-I**{monitor} **Time.relative-B**{da}
Time.relative-I{questa} **Time.relative-I**{mattina}
HardwareOperation.negate-B{non}
HardwareOperation.negate-I{riesco} **null**{piu'}
HardwareOperation.operationType-B{ad}
HardwareOperation.operationType-I{accenderlo}

From this representation attribute names can be easily reconstructed and attribute values can be extracted.
In the remainder of the paper we will evaluate the SLU subtasks of concept segmentation, labeling and value extraction in the context of reranking frameworks.

2) Normalization and Value Extraction: Once a label (concept or attribute) has been assigned by an automatic model, also the attribute values, corresponding to surface forms,

[2]e.g. San Diego is mapped into San-Diego

have to be assigned. Thus, an additional step after concept labeling is the normalization and value extraction. In the LUNA example a possible attribute-value interpretation would be:

HardwareProblem.type[generic_problem] **Peripheral.type**[screen]
Time.relative[morning] **HardwareOperation.negate**[non]
HardwareOperation.operationType[turn_on]

This is the so-called flat attribute-value annotation output by the SLU module. Note that at this level the **null** tags are removed since they are used to annotate words not relevant for the task and so they bring no semantic information. The extracted values are normalized word surface forms, i.e. keywords, used for each concept (in some cases words are also converted into digits).

The are two solutions that can be used to perform the value extraction phase of the SLU task:

(a) rule-based approaches, such as Regular Expressions (RE) to map the words realizing a concept into the corresponding value. These are defined for each attribute-value pair: given a concept and its realizing surface, if a grammar rule, associated with the target concept, matches such surface the corresponding value is returned.

(b) Probabilistic models, which learn the conditional probability $P(V|W, C)$ from manually annotated data, where V is a value, C is a concept and W is the sequence of words.

For the example in the Italian LUNA corpus shown above, where hardware objects are defined, a set of possible surfaces that can be mapped to the concept "Peripheral" is: (i) *screen*; (ii) *the screen*; (iii) *with my screen ...*

Note that all these surfaces share the same keyword, i.e. "*screen*", which can be chosen as a value. All the results in this paper were obtained with approach (a) although some preliminary experiments reveal that (b) is promising and to our knowledge novel. In more detail, the approach (a) must cope with rules that can be in general ambiguous: more than one rule can be applied to the same surface form to extract different values, although such rules are a small subset. Indeed, applying only unambiguous rules gives already acceptable results on manual transcriptions. On automatic transcriptions rules are tuned by hand using complex regular expressions and sorted consistently with respect to two parameters: (1) length of the surface instantiating a concept and (2) rule occurrences. Point (1) avoid applying general rules when more specific ones are available (longer surface). A typical example of point (1) is present in MEDIA: surfaces like "festival de la chanson" are applied before "festival" for the concept **event**. Point (2) is used when no other method can be applied: the most frequent rule (in the training set) is applied.

III. Discriminative Reranking based on Kernel Methods

Previous work has shown that the models typically used for SLU, although accurate, cannot easily encode long-distance dependency relations that exist in the observation and label sequences. Indeed, a fully automatic learning approach, which does not a-priori know which are the relevant dependencies, has to include all possible n-grams in the sequences to attempt

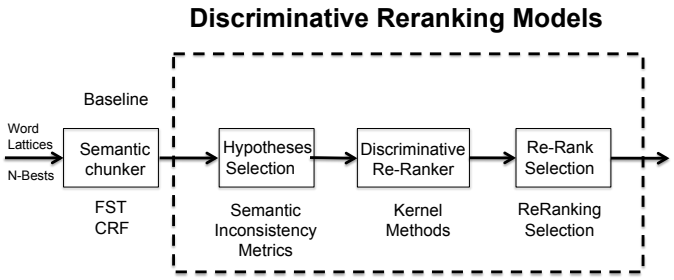


Fig. 1. The DRM computational architecture showing the *fat* pipeline, from speech input to the SLU hypotheses reranking. The ASR module generates n-best or word lattice, which are used as input to a SLU chunker (segmentation and labeling) such as CRF or FSTs. Such hypotheses are used by the DRM module to optimally rerank them using lexical and structural kernels.

capturing all interactions. The number of such n-grams is extremely large[3], consequently, such approach is not practically feasible. Another more practical method is the so-called guided approach, which needs to specify and design the features promising for capturing meaningful relations in advance, e.g. according to a feature-generating scheme based on domain knowledge. Unfortunately, this requires a deep understanding of the language phenomena being studied. A middle approach concerns the approximation of the required feature space by automatically looking for promising dependency features, e.g. [6], [27]. This is very interesting but, being an approximation, it may not generate all the required features.

Our approach implements exhaustive feature generation by exploiting kernel methods, which allows for including all possible dependency features in SVMs. Most importantly, we also propose features capturing the dependency between concepts and standard features (such as words and morphology features). In more detail, we represent the dependency in the conceptual structure by means of semantic trees and sequences that we designed. Then, we apply tree and sequence kernels defined in [16], [18], [20] for extracting all possible substructures, which correspond to different semantic/syntactic dependency features. It should be noted that ours is the first comprehensive study on using such rich semantic features for SLU.

Since the conceptual annotation is needed to capture meaning structures at learning and classification time, we approach SLU by reranking hypotheses, e.g. those provided by local models. This approach is preferable to structural methods, e.g. [28], as their efficient use with tree and string kernels is an open issue. Our DRMs are essentially classifiers of hypotheses pairs $\langle H_i, H_j \rangle$, where H_i, H_j are included in the n-best list (extracted from the hypothesis lattice). These classifiers learn if H_i is more accurate than H_j and, for this purpose, they exploit the whole utterance transcription annotation. This is encoded by our conceptual structures, which are processed by structural kernels.

In the following sections, we describe the baseline models used to generate the semantic hypotheses, the reranking model based on SVMs and the tree-structured features used to represent the hypotheses above. Finallly, we describe two

[3]for example for CRF models the number of features is exponential in the length of the label history (see [2])

enhancements for discriminative reranking: the semantic inconsistency metric and the rerank selection strategy.

A. Baseline Models

The preliminary set of hypotheses of the utterance labeling can be produced by any SLU approach of any complexity, e.g. the model proposed in [27], which already provides a set of dependency features. However, starting from simpler methods offers more appealing advantages: (i) the results are easily reproducible by other researchers and (ii) in case of the corpora we consider, such basic approaches are also the state-of-the-art.

Following this idea, we used two different approaches: (a) generative models, whose probability distributions typically estimate the joint probability of words and (shallow) parses; and (b) discriminative models, which learn a classification function from words to concepts by minimizing the training set error.

In particular, we adopted the generative model based on weighted Finite State Transducers (FSTs), which instantiate SLU as a translation process from words to concepts. This model has shown high accuracy despite its simplicity [7]. One interesting aspect is its easy integration with computational architecture of automatic speech recognition systems, where the output can be word lattices encoded as a weighted FSTs.

Additionally, we used a recent approach for SLU based on CRF [2]. These are undirected graphical models, which achieve state-of-the-art in SLU (e.g. for MEDIA and LUNA). Their training is based on conditional probabilities taking into account many features of the input sequence.

B. Reranker model

Our reranking framework is the one designed in [11]–[13] and detailed in Figure 1: first an ASR outputs a speech transcription, which will be the input of the baseline SLU models. Alternatively, manual transcription can be utilized to study directly the performance of SLU models, without the negative impact of the ASR in the overall pipeline.

Second, the baseline SLU model, in our case the SFST or the CRF, takes the transcription of a spoken sentence as input and produces the *n* most likely conceptual annotations for the sentence. These are ranked by the joint probability of the Stochastic Conceptual Language Model (SCLM) in case of SFST or by the global conditional probability of the concept sequence given the input word sequence when CRF are used. The *n*-best list produced by the baseline model is the list of candidate hypotheses, e.g. $H_1, H_2, .., H_n$, used in the next reranking step.

Third, the SIM module evaluates and selects the semantic consistency of ASR hypotheses. This processing step is described in Section III-E and it is used to improve the quality of the *n*-best list.

Next, the produced hypotheses are used to build pairs, e.g. $\langle H_1, H_2\rangle$ or $\langle H_1, H_3\rangle$. We build training pairs such that a reranker can learn to select the best between two hypotheses of a pair, i.e. the hypothesis containing the least number of mistakes with respect to a reference metric. Such classifier can be applied to provide the final ranked list.

Finally, the confidence of the reranker, i.e. the SVM score, can be optionally compared with the one of the basic SLU model to select the most reliable output (RRS). Hereafter, we provide more details on the training of our rerankers.

1) Reranker training and classification: Given the following two annotations of the input sentence "*ho un problema col monitor*" ("*I have a problem with the screen*"):

H_1: ***NULL*** *ho* ***PROBLEM-B*** *un* ***PROBLEM-I*** *problema* ***HARDWARE-B*** *col* ***HARDWARE-I*** *monitor*

H_2: ***NULL*** *ho* ***ACTION-B*** *un* ***ACTION-I*** *problema* ***HARDWARE-B*** *col* ***HARDWARE-B*** *monitor*

we build the pair $\langle H_1, H_2\rangle$, where **NULL**, **ACTION** and **HARDWARE** are the assigned domain concepts. A pair is a positive training instance if the first hypothesis (H_1 in the example) has a lower concept annotation error rate than the second (H_2), with respect to the reference manual annotation, and negative otherwise. In our example, the second annotation is less accurate than the first since *problema* is erroneously annotated as ***ACTION*** and *"col monitor"* is erroneously split in two different concepts.

In order to effectively train the reranker, we proceed as follows: first, we select the best annotation H_k in the n-best list by measuring the edit distance of all hypotheses with respect to the manual annotation; second, we generate the positive instances as pairs $\langle H_k, H_i\rangle$, for $i \in [1..n]$ and $i \neq k$, and negative instances as pairs $\langle H_i, H_k\rangle$. At classification time, since we cannot compare hypotheses with the reference, all possible pairs $\langle H_i, H_j\rangle$, with $i, j \in [1..n]$ and $i \neq j$, must be generated. Nevertheless, using the simplification described in [29], we can use single hypotheses instead of pairs[4], thus the classification instances are only n, instead of n^2. This simplification is based on the fact that, as pairs for the training phase are symmetric, the final model can be represented as a hyperplane passing through the origin of coordinates, thus also at classification phase, the score of a pair $\langle H_i, H_j\rangle$ is the opposite of the symmetric pair $\langle H_j, H_i\rangle$.

C. The Reranking Kernel

We adopt the kernel introduced in [30] for preference ranking with ordinal regression and used in [29] for parse tree reranking and in [17], [19] for predicate argument structure reranking. Given the definition of a generic pair of hypotheses $e_l = \langle H_{l,1}, H_{l,2}\rangle$, the kernel applied to two pairs e_1, e_2 computes:

$$\begin{aligned} K_R(e_1, e_2) &= K(H_{1,1}, H_{2,1}) + K(H_{1,2}, H_{2,2}) \\ &- K(H_{1,1}, H_{2,2}) - K(H_{1,2}, H_{2,1}), \quad (1) \end{aligned}$$

where K can be any kernel function, for example those described in [15], [18], i.e. String Kernel (SK), Syntactic Tree Kernel (STK) and Partial Tree Kernel (PTK).

It is worth noting that: first, our reranking schema, consisting in summing four different kernels, has been already applied in [29], [31] for syntactic parsing reranking, where the basic kernel was a Tree Kernel.

Second, in [32], an equivalent reranking model was applied to different candidate hypotheses for machine translation, but

[4]More precisely, a pair with only one hypothesis, i.e. $\langle H_i, \emptyset\rangle$

the goal was different and, in general, simpler: our task consists in selecting the best annotation of a given input sentence, while in [32], the task is to distinguish between *"good"* and *"bad"* translations of the same sentence.

Third, the reranking approach brings several advantages, but also some disadvantages as the reranker training time is affected by the one of SVMs, which are trained with a quadratic programming algorithm. However, there are very fast approaches [33], [34] and methods transforming structural kernels in linear kernels [35], for which linear training and testing algorithms exist (e.g. the cutting-plane algorithm).

Finally, two main advantages can be observed in the reranking model. The first is the ability to put together characteristics of two different models. The second is that using kernels like String and Tree Kernels, the reranking model can capture arbitrarily long-distance dependencies between words and concepts using the whole semantic annotation generated by the baseline model. In contrast, the basic models described in this work can capture dependencies only between words, or word features, and concepts at a limited distance: trigrams for the SFST model, bigram for CRF. The latter can reach in any case high accuracy since it can use many features of the input sequence and learns directly global posterior probabilities.

D. Structural features for reranking

The kernels described in [14]–[20] provide a powerful technology to capture structured features from data, but the latter should be adequately represented. We propose the following two sequential structures, $SK1$ and $SK2$, to represent SLU hypotheses in the sequence kernel (SK) defined in [18]:

$SK1$ ***NULL*** *ho* ***PROBLEM-B*** *un* ***PROBLEM-I*** *problema* ***HARDWARE-B*** *col* ***HARDWARE-I*** *monitor*

$SK2$ ***NULL*** *ho* ***PROBLEM B*** *un* ***PROBLEM I*** *problema* ***HARDWARE B*** *col* ***HARDWARE I*** *monitor*,

where the B/I tags characterize the *Begin* and *Inside* (or continuation) of multiword concepts, as described also earlier. For both SK1 and SK2, the order of words and concepts is meaningful since each word is preceded by its corresponding concepts, so a generic sequence $\texttt{concept}_i$ $\texttt{word}_j$ captures a dependence between i and j while the sequence $\texttt{word}_j$ $\texttt{concept}_i$ does not. Also note that SK1 is more precise than SK2 since it links the B/I tags together with the concept, but at the same time, it is more sparse since it produces a larger number of labels.

The above representation is powerful since can capture all possible dependencies but it is also rather flat. Therefore, to better exploit the power of kernels, we build tree-like structures directly from semantic annotation. Note that the latter is made upon sentence chunks, which implicitly define syntactic structures as long as the annotation is consistent in the corpus. This way we do not need to use syntactic parse trees and augment them with domain specific information, e.g. semantic tags. In more detail, we propose the structures for tree kernel processing shown in Figure 2(a), 2(b) and 3, where the semantic tree in the latter figure along with STK and PTK (see [18]) allows for generating a wide number of features (like Word categories, POS tags, morpho-syntactic features), which are commonly used in this kind of tasks.

Moreover, we point out that: (a) we only use Word Categories as features in the semantic trees. Such categories can be domain independent like *"Months"*, *"Dates"*, *"Number"* etc., or a POS-tag subset (used to generalize target word prefixes in inflexive languages) such as Articles, Prepositions, Possessive and Demonstrative Adjectives. (b) The features in common between two trees must appear in the same child-position, hence they are sorted based on feature indexes, e.g. *F0* for words and *F1* for word categories.

Note that the proposed semantic structures shaped as trees only encode the information pertinent to the task, i.e., the concepts annotated in a given sentence, their segmentation in chunks, the surface form of each concepts and some features needed to improve generalization. In contrast, structures built on top of syntactic parse trees would be very large and may contain information, often not needed. Thus a fine pruning of them would be needed in order to make them effective.

E. Hypothesis Selection Criteria via Inconsistency Metrics

An interesting strategy to improve reranking performance is a pre-selection of the best set of hypotheses to be reranked. In previous work [11]–[13], [29], [31], [32], [36], no study in this direction has been carried out, i.e. the *n*-best hypotheses generated by the baseline model were used for reranking.

In this work we propose a Semantic Inconsistency Metric (SIM) based on the attribute-value extraction (AVE) step of the SLU process that allows for selecting better hypotheses used afterwards in the reranking phase.

The attribute-value extraction module is based on rules that map words (or word sequences) into the corresponding value. For this purpose, the conceptual information annotated by the baseline model is also used.

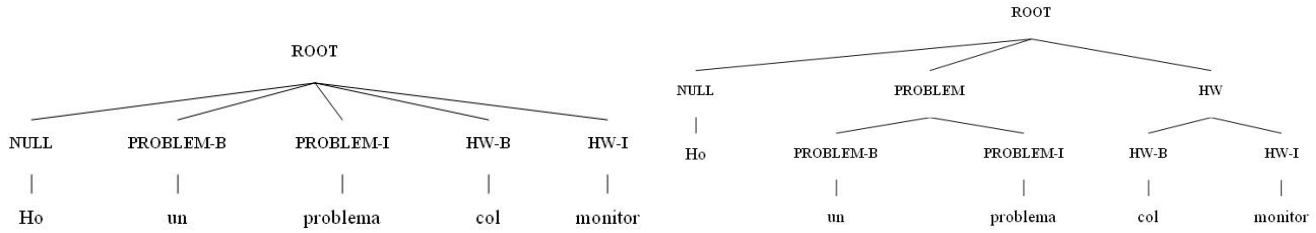
The rules are defined to extract values from well formed phrases annotated with correct concepts. Thus, when the corresponding words are annotated with a wrong concept by the baseline model, the extracted value will probably result wrong. We use this property to compute a semantic inconsistency value for the hypotheses, which allows us to select better hypotheses, i.e. with higher probability to be correct.

We show our SIM using the same example already used before, where hypotheses are produced starting from the sentence "*I have a problem with my screen*". From it, three possible hypotheses may be generated by the baseline model, where we suppose to have already removed the **Null** concept associated with the chunk "*I have*":

1) **Action**{*a problem*} **Peripheral**{*with my screen*}
2) **Problem**{*a problem*} **Peripheral**{*with my screen*}
3) **Problem**{*a problem*} **Peripheral**{*with my*} **Peripheral**{*screen*}

Two of these annotations show typical errors of an SLU model:
(i) wrong concepts annotation: in the first hypothesis the phrase "a problem" is erroneously annotated as **Action**;
(ii) wrong concept segmentation: in the third hypothesis, the phrase "with my screen" is split in two concepts.
If we apply the AVE module to these hypotheses the result is:

1) **Action**[] **Peripheral**[screen]
2) **Problem**[general_problem] **Peripheral**[screen]



(a) *FLAT* Tree (b) *MULTILEVEL* Tree

Fig. 2. Examples of "*FLAT*" and "*MULTILEVEL*" semantic trees used for STK and PTK

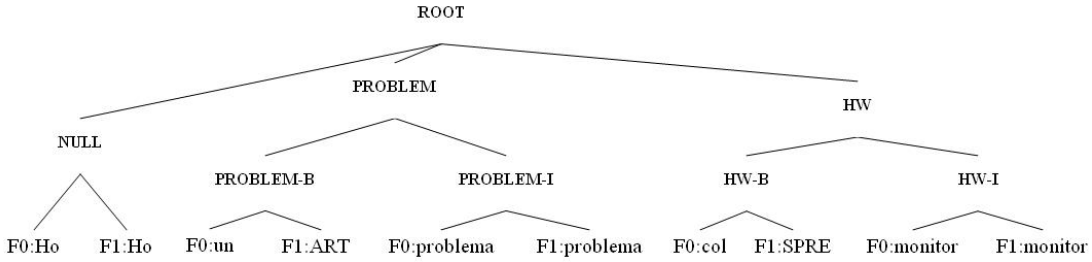


Fig. 3. An example of "*FEATURES*" semantic tree used for STK or PTK

3) **Problem**[general_problem] **Peripheral**[] **Peripheral**[screen]

We note that **Action** has an empty value since it was incorrectly annotated and, therefore, it is not supported by words from which the AVE module can extract a correct value. In this case, the output of AVE can only be empty. Similarly, for the third hypothesis, the AVE module cannot extract a correct value from the phrase "with the" since it doesn't contain any keyword for a **Peripheral** concept.

For each hypothesis, our SIM counts the number of possibly wrong values, i.e. empty values. In the example above, we have 1, 0 and 1 for the three hypotheses, respectively. Accordingly, the most accurate hypothesis under SIM is the second, which is also the correct one in this case.

We exploit SIM by generating a huge number of hypotheses with the baseline model and selecting only the top *n*-best with respect to the SIM score. These hypotheses are then used in the discriminative reranking step. Such strategy gives the advantage of choosing hypotheses from a large set, where it is probable to find a more correct annotation. In contrast, all the previous reranking approaches directly used the raw *n*-best list provided by baseline model. Moreover, in order to limit computational cost, the size of the *n*-best list is kept relatively small (few tens in the best case).

F. Rerank Selection (RRS)

A reranking model can generally improve the baseline model used as hypotheses generator. The intuition behind this claim is that a reranker can infer the statistical distribution of the baseline model mistakes. Moreover, for this purpose, it can use the semantic annotation and its consistency over the whole input sentence, i.e. it can use features capturing statistical dependencies spanning the whole hypothesis. On the other hand, a reranker is a statistical classifier, which is subject to errors with some probability. This means that for some input sentences, the top ranked hypothesis can be less accurate than the original best one provided by the baseline model.

We can exploit the above consideration to further improve the reranking framework; we can build meta-classifiers that, using meta-features, choose between the outcome of the reranker and the baseline model. One simple and efficient way to design such meta-classifier is the use of the classification scores of the two competing systems above and select the most reliable one: we call this approach '*ReRank Selection* (RRS). In more detail, it requires the estimation, with respect to error rate minimization, of two confidence thresholds applied to the scores of the baseline and the reranking model. Given such optimal thresholds, we choose the final best SLU hypothesis with the following decision function:

$$BestHypothesis = \begin{cases} HYP_{RR} \; if(C_{fst} \leq T_{fst} \; and \; C_{RR} \geq T_{RR}) \\ HYP_{fst} \qquad otherwise \end{cases}$$

where HYP_{RR} and $HYP_{fst/crf}$ are the best hypotheses derived by the reranker and the baseline model (SFST or CRF) with their associated score C_{RR} and $C_{fst/crf}$, respectively. T_{RR} and $T_{fst/crf}$ are the two thresholds trained for the decision function.

It should be noted that: (i) we use two thresholds in our models since the two system scores cannot be directly compared: the SVM outcome is a functional margin whereas CRF is a probability. Combining these two scores would require a scaling parameter for one of them in order to give the correct weight. This in turn would require optimization of such scaling parameter. (ii) The two thresholds provide more robustness since they set a specific reliability limit for each classifier. (iii) This meta-model, although simple, is effective in exploiting errors made by the baseline and the reranker, since it uses both scores for prediction, while the reranking model can only exploit the baseline model score.

IV. EXPERIMENTS

The aim of the experiments is to show that our DRMs can effectively exploit rich dependency features for improving state-of-the-art in SLU (at least with respect to our referring

corpora). For this purpose, we first carried out experiments to study the impact of different kernels and structures on the reranking accuracy of baseline models. Secondly, we compare the reranking algorithms against state-of-the-art models, i.e. FST, SVMs and CRF. We use benchmarks that aim at representing tasks of different complexity and language variability, i.e. the ATIS [21], MEDIA [22] and LUNA [10] corpora. Third, to have real scenario results, we also compare models on automatic transcriptions generated by Automatic Speech Recognition systems (ASR).

Finally, we present the results of our DRM computational architecture, shown in Figure 1, which exploits the inconsistency metrics for hypothesis pre-selection and the reranking model selection strategy (to activate or deactivate DRMs).

A. Experimental setup

All the SCLMs that we apply in the experiments either for the FST model baseline or to produce the input for the reranking model, are trained with the SRILM toolkit [37] using an interpolated model for probability estimation with the Kneser-Ney discount [38]. We then converted the model in an FST again with SRILM toolkit. One of the drawback of such model is that its accuracy is affected by the presence of Out-of-Vocabulary words (OOV). We have solved this problem by mapping words into word categories, which are usually not OOV. For ATIS and MEDIA corpora, word categories were provided together with the corpus, while for the Italian LUNA corpus categories have been designed together with the ontology used for annotating the corpus (more details are given in [23]). Thus, in all cases word categories are part of the application knowledge base. For example, city names can be grouped in the category **CITY**: if a city name, e.g. *Bordeaux*, does not appear in the training set, we can back-off to CITY category, which accounts for other cities, e.g. *Paris*, appearing in the training set. Since the FST model first maps words into categories, *Bordeaux* is mapped into the category **CITY**. This simple solution gives the possibility to correctly tag also OOV words. The OOV problem is still present but affect much less the model performance.

The SVM baseline for concept classification was trained using YamCHA[5] [39]. The CRF models were trained with the CRF++ tool.[6] The parameter settings are described in [40], which is the state-of-the-art on the corpora we consider. Indeed in [40], CRF are compared with other four models (SFST, SVMs, Machine Translation, Positional-Based Log-linear model) by showing that they are by far the best models on the MEDIA corpus. We used the same features for both SVM and CRF baseline, i.e. word and morpho-syntactic features in a window of [-2, +2] with respect to the current token, plus bigrams of concept tags (see YamCHA and CRF++ web site and [40] for more details).

The reranking models based on structured kernels and SVMs were trained using the SVM-Light-TK toolkit.[7] The number of hypotheses used for reranking was always set to 10. The larger is the number of hypotheses the larger will be the oracle accuracy, but we have to trade-off the latter with efficiency. The SIM algorithm (Sec. III-E) selects 10 out of 1,000 hypotheses from the baseline model (the large number of rejected hypotheses did not contribute to the oracle accuracy of the system).

The thresholds for the decision function of the RRS strategy (see Sec. III-F) are trained on the development set of the corresponding corpus.

Our approach for training the reranking models is called *"Split Training"* (ST) in [12], and it has been used in many works about reranking, e.g. in [31]. We split the training set in two parts: a first FST model is trained on part 1 and generates the *10*-best hypotheses parsing part 2, thus providing the first chunk of reranker's data. Then the same procedure is applied inverting part 1 with part 2 to provide the second data chunk. Finally, the reranker is trained on the merged data. For classification, the *10*-best hypotheses of the entire test set are generated using the FST model trained on all training data.

For the ATIS experiments, we did not apply any parameter optimization, i.e. we used the parameters from previous work. For the experiments on MEDIA and the Italian corpora, we optimized all the parameters on the development sets.

The results are expressed in terms of concept error rate (CER). This is a standard measure based on the Levensthein alignment of sentences and it is computed as the ratio between inserted, deleted and confused concepts and the number of concepts in the reference sentence. When not specified, CER is computed only on attribute names (**Attr.**), otherwise CER is computed for both attribute names and values (**Attr-Value**).

Since we also tested the SLU models on automatic transcriptions, we report that the latter were produced by a speech recognizer with a WER of 10.4%, 27.0% and 31.4% on the ATIS, LUNA and MEDIA test sets, respectively. In all cases the used language model is an interpolated model with Kneser-Ney discount [38], which gives a better performance in most cases.

1) Training and Classification Time Issues: All models described in this work have been trained on machines with two CPUs Xeon dual-core 2.3 GHz and 4 or 8 GB of RAM. We report training time on MEDIA since it is the largest corpus: using CRF++, we had to use features cut-off (with a threshold of 2) in order to be able to fit data into the central memory. Even in this setting the training time was roughly 5 days, to which the training time for the reranker has to be added. This was between 7 and 9 days, depending on the structure used for the kernels.

Higher memory availability allows for using more features with CRF (without decreasing training time) and increasing the kernel cache size for the reranker, which significantly increases the speed of kernel computations. For our latest experiments, we used machines with 64GB of RAM (and same computational power as before), which resulted in a training time for reranking models of roughly 2.5 days.

Concerning classification time, all baseline models, including CRF, are fast enough to be used in real time applications. For example, the CRF model for MEDIA can generate hypotheses for a sentence in roughly 0.6 seconds. In contrast the

[5]available at http://chasen.org/∼taku/software/yamcha

[6]available at http://crfpp.sourceforge.net/ (from the same author of YamCHA)

[7]available at htttp://disi.unitn.it/moschitti

Structure	STK	PTK	SK
FLAT	18.5	19.3	-
MULTILEVEL	20.6	19.1	-
FEATURES	19.9	18.4	-
SK1	-	-	16.2
SK2	-	-	18.5

TABLE IV
CER OF RERANKERS USING STK, PTK AND SK ON LUNA (MANUAL TRANSCRIPTIONS) APPLIED TO THE FSTS' OUTPUT. FSTS AND SVMS ACHIEVE A CER OF **23.2%** AND **21.0%**, RESPECTIVELY.

reranking model evaluates all the hypotheses in roughly 11 seconds per sentence. This time is rather high for real time applications, nevertheless SVM classification can be easily parallelized.

B. Comparing Kernels and Semantic Structures

Table IV shows the accuracy of rerankers using different kernels applied to different semantic structures. Such results refer to our previous work in [11], for which we used an older version of the LUNA corpus (a subset of the corpus used in this work). We exploit this outcome to motivate our choice of the best combination of kernels and structures to be used for the comparison against the state-of-the-art. The dash symbol appears where the kernel cannot be applied to the corresponding structure.

It is worth noting that: first, from Table IV, rerankers significantly improve the baseline results, i.e. 23.2% (CER for FST) and 21.0% (CER for SVMs). For example, SVM reranker using SK, in the best case, improves FST concept classifier of 23.2-16.2 = 7 points.

Second, the structures designed for trees yield rather different results depending on the used kernel. We can see in Table IV that the best result using STK is obtained with the simplest structure, i.e. *FLAT*, while with PTK the best result is achieved with the most complex structure, i.e. *FEATURES*. This is due to the fact that STK does not split the children of each node, as explained in [15], and so structures like *MULTILEVEL* and *FEATURES* result too rigid and prevent STK to be effective. In contrast, the structure *FLAT* is rigid as well, but since it is very simple and has only one level of nodes it can capture the most meaningful features.

Third, we do not report all the results using different kernels and structures for the MEDIA corpus. However, we point out that since MEDIA is a noticeable larger corpus and its processing is also more complex (42 concepts in LUNA, 99 in MEDIA), the more complex structures are also more effective to capture word-concept dependencies.

Finally, the String Kernel applied to the structure *SK1* seems to be the most effective with a CER of 16.2% on the first and smaller version of LUNA. However, since SK is computationally demanding, we cannot apply it to large corpora, e.g. MEDIA or even ATIS. Moreover, in the next section an experiment on the new version of the LUNA corpus will show that SK is not more accurate than PTK. For these reasons, we adopted the PTK with the richest tree structure *FEATURES* in all the following reranking experiments: this is the best trade-off between accuracy and computational complexity. We used such settings in all the following experiments on the MEDIA and the Italian LUNA corpora and for both FST and CRF reranking.

Model	$ATIS_{man}$ (CER)	$ATIS_{auto}$ (CER)
FST	6.7%	13.5%
SVM	6.9%	13.8%
CRF	7.1%	14.0%
FST+RR (PTK)	6.2%	13.2%

TABLE V
RESULTS OF SLU EXPERIMENTS ON THE ATIS CORPUS USING MANUAL ($ATIS_{man}$) AND AUTOMATIC TRANSCRIPTIONS ($ATIS_{auto}$) WITH A WORD ERROR RATE (WER) OF THE ASR OF 10.4%.

ATIS concepts	counts
ArrivalCity	5043
DepartureCity	5018
DepartureDate.day_name	1100
AirlineName	802
DepartureTime.period_of_day	683
DepartDate.day_number	450
DepartDate.month_name	435
DepartTime.time	426
RoundTrip	421
DepartTime.time_relative	387

TABLE VI
TOP MOST OCCURRING CONCEPTS IN THE ATIS CORPUS.

Regarding the use of kernels an interesting finding can be derived: kernels producing a high number of features, e.g. SK or PTK, in general produce higher accuracy than kernels less rich in terms of features, e.g. STK. In particular STK is improved by 2.3 percent points (Table IV).

C. Cross-Corpus Comparisons using Basic Rerankers

In these experiments, we used the combination of PTK with the structure FEATURES to design our reranker as it provides the best compromise between accuracy and efficiency, according to the previous section. We compare it across different corpora, i.e. ATIS, MEDIA and LUNA, respectively.

Table V reports the results on ATIS, obtained with the same setting of the three baseline models (FST, SVM and CRF). Since FST results the best model, we only compare with the reranker built on top of the FST model.

ATIS is the simplest task and this is reflected in high accuracy for all models, even using automatic transcriptions coming from an ASR system. Nevertheless it is worth discussing some interesting outcomes. The errors made on the ATIS test set are caused by an imbalanced amount of instances of concepts. Indeed, Table VI shows that the concepts **DepartureCity**, **ArrivalCity** and **DepartureDate.day_name** are by far the most frequent (57.7% of the total counts). This means, for instance, that the models are strongly biased to annotate a city as *Departure* or *Arrival* city, regardless what the context is. Note that the reranking model, FST+RR (PTK), even in this situation, can improve individual systems. The improvement is only 0.5% points on manual transcriptions, with respect to the baseline FST model, since the FST model error rate is very small.

Note that on ATIS there is no value extraction phase since values basically correspond to surfaces realizing each concept. Thus, the values for this task are obtained by simple and deterministic processing of surface forms (the ATIS corpus used for this task is the same used in [24] and [25]). For this reason, we have judged not worthwhile applying the improved reranking models (described in following sections) to ATIS.

Tables VII and VIII show results of the SLU experiments on the MEDIA and LUNA test sets using manual and automatic

Model	MEDIA(CER)		LUNA IT(CER)	
	Attr.	Attr.-Value	Attr.	Attr.-Value
FST	14.2%	17.0%	24.4%	27.4%
SVM	13.4%	15.9%	25.3%	27.1%
CRF	11.7%	14.2%	21.3%	23.5%
FST+RR	11.9%	14.6%	20.6%	23.1%
CRF+RR	11.5%	14.1%	19.9%	21.9%
CRF+RR$_{SK}$	-	-	21.1%	23.4%

TABLE VII
RESULTS OF SLU EXPERIMENTS ON THE MEDIA AND THE ITALIAN LUNA TEST SETS ON MANUAL TRANSCRIPTIONS. SK INDICATES THE USE OF SK INSTEAD OF THE USUAL PTK.

Model	MEDIA(CER)		LUNA IT(CER)	
	Attr.	Attr.-Value	Attr.	Attr.-Value
FST	28.9%	33.6%	36.4%	39.9%
SVM	25.8%	29.7%	34.0%	36.7%
CRF	24.3%	28.2%	31.0%	34.2%
FST+RR	25.4%	29.9%	32.7%	36.2%
CRF+RR	23.6%	27.2%	29.0%	32.2%

TABLE VIII
RESULTS OF SLU EXPERIMENTS ON THE MEDIA AND THE ITALIAN LUNA TEST SETS ON AUTOMATIC TRANSCRIPTIONS (ASR WER IS 31.4% FOR MEDIA AND 27.0% FOR LUNA)

transcriptions of spoken sentences, respectively. In these tables we compare all baseline models (FST, SVM and CRF) and the reranking models based on FST and CRF hypotheses (FST+RR and CRF+RR).

As we can see from these tables the most accurate baseline model is CRF. This is not surprising since we replicate the CRF models that showed the best performance on some SLU tasks as described also in [40]. It is worth noting that the two reranking models proposed in this work improve their respective baseline models. For instance, FST+RR improves the FST baseline of 2.4% and 3.7% on MEDIA and LUNA corpora, respectively, on attribute-values extraction from manual transcriptions (text input from now on). For automatic transcriptions (speech input from now on) the improvement is of 3.7% for both corpora.

In contrast, although CRF+RR still improves the CRF baseline, the improvement is much smaller, i.e. 0.1% on MEDIA but still meaningful on LUNA, i.e. 1.6% for text input and attribute-values extraction. This is due to the higher accuracy of CRF on MEDIA, which leaves much less improvement margin. This intuition is confirmed by the results of CRF+RR model on speech input, where, since the baseline CER is rather higher, the improvement is significant. Still considering the same tasks as above, i.e. MEDIA and LUNA corpora and attribute-values extraction, the gain in CER is 1.0% and 1.6% respectively.

Finally, the last row of the Table VII reports the CER of reranking using SK, which is higher than the one produced by PTK. This confirms that the choice of the latter is the most appropriate. Regarding the very high result obtained by SK in Table IV, we found out that it is due to the particular characteristics of the first version of the LUNA corpus (e.g. rather small, more noisy and less number of concepts) used in such experiments.

D. Cross-Corpus Comparisons Using Enhanced Rerankers

In these experiments, we applied two enhancements of DRMs: the SIM and RRS strategy. Tables IX shows comparative results on text input between FST, SVM and CRF against RRS (described in Section III-F). We note that RRS improves accuracy of both FST and CRF. Although, in some cases the gain is small 0.4% = 11.7% - 11.3% for CRF in the worst case, i.e. on MEDIA, attribute extraction and text input, there is a constant improvement in all tasks (with a maximum gain of 4.6% = 27.4% - 22.8% wrt to FST on LUNA for attribute-values extraction).

Model	MEDIA(CER)		LUNA IT(CER)	
	Attr.	Attr.-Value	Attr.	Attr.-Value
FST+RRS	11.7%	14.3%	20.7%	22.8%
CRF+RRS	11.3%	13.6%	19.9%	21.9%

TABLE IX
RESULTS OF SLU EXPERIMENTS ON MEDIA AND ITALIAN LUNA TEST SETS ON MANUAL TRANSCRIPTIONS USING RE-RANK SELECTION.

Model	MEDIA(CER)		LUNA IT(CER)	
	Attr.	Attr.-Value	Attr.	Attr.-Value
FST+RRS	25.0%	29.2%	31.8%	35.5%
CRF+RRS	23.2%	26.8%	28.8%	31.9%

TABLE X
RESULTS OF SLU EXPERIMENTS ON MEDIA AND ITALIAN LUNA TEST SETS ON AUTOMATIC TRANSCRIPTIONS USING RE-RANK SELECTION.

More interesting results can be observed on speech input, shown in Table X, where the minimal improvement over CRF is 1.1% = 24.3% - 23.2% and 1.4% = 28.2% - 26.8% on MEDIA (attribute and attribute value extraction, respectively) and the maximum improvement over FST is 6.0% = 36.4% - 30.4% and 6.1% = 39.9% - 33.8% on LUNA.

Finally, tables XI and XII report a final comparison of all our reranking models, also including the hypothesis selection strategy (SIM), on LUNA and MEDIA corpora, respectively.

We note that the best DRMs, which use RRS and SIM, i.e. FST+RRS+SIM or CRF+RRS+SIM, significantly improve the other rerankers. For example, FST+RRS+SIM improves the model FST+RRS of .4% in the worst case (MEDIA text input) and 1.7% in the best case (LUNA speech input). Similar improvement is achieved by the model CRF+RRS+SIM on CRF+RRS (0.2% in the worst case, 0.5% in the best case). These results suggest that our simple hypothesis selection constantly improves DRM. Indeed, it allows for selecting hypotheses from a larger set than a simple reranking model, where just 10-20 hypotheses are considered (e.g. see [31]).

The overall enhancement on CRF, which is the best model, is: 2.3%, 2.4% and 2.7% and 2.8% on LUNA, text input (Attr. and Attr. values) and speech input (Attr. and Attr. values), respectively. The improvement on MEDIA is 0.6%, 1.1% and 1.6%, 1.9%, text and speech input respectively.

It should be noted that:

• these results are important since they improve on the state-of-the-art (reached by CRF). For example, for attribute recognition on manual transcribed data, the best CER reported on MEDIA is 11.5 [40], which is comparable with our baseline, i.e. 11.7. In the paper above, better results than the latter are also reported but they refer to different improved implementations of the CRF training algorithm thus not related to feature representation.

• Reranking may be limited by the quality of the hypotheses generated by the local model. To show that this is not an important limitation, in Table XIV, we report the Oracle Error Rates of our rerankers on all three corpora used for our

	Text Input (CER)		Speech Input (CER)	
Model	**Attr.**	**Attr.-Val.**	**Attr.**	**Attr.-Val.**
FST+RRS+SIM	19.2%	21.5%	30.4%	33.8%
CRF+RRS+SIM	19.0%	21.1%	28.3%	31.4%

TABLE XI
RESULTS ON LUNA CORPUS USING BOTH MANUAL TRANSCRIPTIONS (TEXT INPUT) AND AUTOMATIC TRANSCRIPTIONS (SPEECH INPUT).

	Text Input (CER)		Speech Input (CER)	
Model	**Attr.**	**Attr.-Val.**	**Attr.**	**Attr.-Val.**
FST+RRS+SIM	11.3%	13.8%	24.5%	28.2%
CRF+RRS+SIM	11.1%	13.1%	22.7%	26.3%

TABLE XII
RESULTS ON MEDIA CORPUS USING BOTH MANUAL TRANSCRIPTIONS (TEXT INPUT) AND AUTOMATIC TRANSCRIPTIONS (SPEECH INPUT).

experiments in this work.

• These show that there is a large gap with respect to the current best results and there is a large margin of improvement using our DRMs.

1) Insight on SIM: The improvement of SIM on our DRMs makes its investigation worthwhile, especially with respect to its impact on the selection of hypotheses before the use of the reranker. This can be evaluated by testing the accuracy of the baseline model after applying SIM alone. The CER of SIM applied to the CRF n-best list (CRF+SIM) on manual transcriptions is reported in the first row of Table XIII, for both LUNA and MEDIA corpora. We note that SIM slightly improves CRF, i.e. 0.5% and 0.7% on LUNA and MEDIA for attribute-value extraction, respectively (compared with Table VIII).

It is also interesting to test how the oracle accuracy of the hypotheses changes after SIM (Oracle and Oracle$_{SIM}$). The oracle CER is computed by measuring the edit distance between each hypothesis and the manual annotation and taking the one with the least number of mistakes. The improvement of roughly 2.0% (Table XIII, second and third rows) on both LUNA and MEDIA demonstrates the general validity of SIM.

E. Statistical Significance of the Results

Some of the results derived in this paper show slight improvement of one model over the other, which prevents to derive significance of some outcome. For this reason, we evaluated significance tests of all our results on manual transcriptions, for both LUNA and MEDIA corpora. We do not report the same analysis for automatic transcriptions, although the higher difference typically achieved between models for them should guarantee a significance of our results.

For the statistical significance tests, we used the software by Sebastian Pado (available at http://www.nlpado.de/ sebastian/sigf.shtml). This carries out the computationally-intensive randomization test described in [41], which is particularly suitable for measures such as Precision, Recall or F1; we have adapted it for the Concept Error Rate of our models. It tests the following null hypothesis: given two models with performances R_1 and R_2 (in our case R is the Concept Error Rate), the test evaluates how likely is to observe a difference in the results at least as large as $R_1 - R_2$. Since the assumption is that models are equal, if the probability is lower than a certain confidence, we can state that the difference is statistically significant (with respect to such confidence).

	LUNA		MEDIA	
Model	**Attr.**	**Attr.-Val.**	**Attr.**	**Attr.-Val.**
CRF+SIM	21.2%	23.0%	11.5%	13.5%
Oracle	15.5%	18.6%	7.3%	9.5%
Oracle$_{SIM}$	14.4%	16.8%	5.8%	7.5%

TABLE XIII
IMPACT OF SIM ON 10-BEST HYPOTHESES FROM CRF (MANUAL TRASCRIPTION AND NO RERANKING).

CORPUS	**CER** Attr.	**CER** Attr.-Val.
ATIS	3.1%	4.3%
MEDIA	5.8%	7.5%
LUNA	14.4%	16.8%

TABLE XIV
ORACLE CER ON THE ENGLISH ATIS, FRENCH MEDIA AND LUNA ITALIAN CORPORA (MANUAL TRANSCRIPTION).

We report the significance test for a subset of our models in Table XV. Given two models M_1 and M_2, M_1 vs. M_2 is associated with a score of statistical significance, i.e. the p-score indicating statistical significance. We provide the significance test for the most important comparisons as the full set of comparisons would require a combinatorial number of models.

The results show that most of the CER difference between models are statistically significant. The only important exception is CRF+RR vs CRF+RRS on the LUNA corpus. However, this is not completely unexpected as their outcomes are rather similar and LUNA results are also affected by the small size of the data. Additionally, low statistically significant scores are observed for CRF+SIM, i.e. the application of SIM without applying reranking. In summary, the confidence test assesses the validity of our DRMs.

V. DISCUSSION AND CONCLUSIONS

In this section we first summarize the ideas and techniques reported in this paper, then we assess them by discussing the related work and finally we give an outline of the empirical results achieved by our models.

A. Qualitative Analysis

An important characteristic of our tree-shaped structures used by PTK is the ability to capture long distance dependencies. This is confirmed by our comparative analysis between the outcome of the baseline models and our DRMs, performed on the outcome on MEDIA. In more detail, MEDIA contains different concepts providing similar information, which can be only correctly classified by carefully considering their context. For example, *temps-date* (*time-date*) and *temps-jour-mois* (i.e. *time-day-month*) provide similar information about time expressions. The first refers to time expressions used for a hotel reservation whereas the other indicates a general expression of time. The higher frequency of standard time concept (*temps-jour-mois*) biases the prior of the SLU model. Thus, intuitively, when the context cannot be identified by the baseline model, the concept *temps-jour-mois* will be selected. In contrast, DRMs provide much more context through long-distance dependencies (e.g. with other concepts expressed in the sentence like $booking$). It is interesting to show that CRF mistook this concept 13 times, while after reranking the same concept was mistaken only 7 times.

	LUNA		MEDIA	
Model Pair	**Attr.**	**Attr.-Val.**	**Attr.**	**Attr.-Val.**
CRF+RR vs. CRF+RRS	0.3264	0.1007	9.99e-5	0.0025
CRF+RR vs. CRF+RRS+SIM	9.99e-5	9.99e-5	9.99e-5	9.99e-5
CRF+RR vs. CRF+SIM	0.3653	0.2896	0.0064	9.99e-5
CRF+RRS vs. CRF+RRS+SIM	0.0389	0.0017	9.99e-5	9.99e-5
CRF+RRS vs. CRF+SIM	0.0412	0.1997	9.99e-5	9.99e-5
CRF+RRS+SIM vs. CRF+SIM	0.0011	0.0119	0.1349	0.3309

TABLE XV
SIGNIFICANCE TESTS ON THE MOST MEANINGFUL MODELS DESCRIBED IN THIS WORK (THE LOWER THE VALUES THE MORE SIGNIFICANT).

Other similar concepts falling into the same rationale (and so mistaken for the same reason) are: (i) *localisation-lieurelatif-nomme* (*localization-relative-place-name*) and *localisation-lieurelatif-general* (*localization-general-relative-place*): for which the number of errors are 11 and 5 for baseline and the reranking models, respectively; (ii) *sejour-nbnuit* (*journey-number-of-nights*) and *temps-unite* (*time-unit*), mistaken 11 and 6 times, respectively, and *localisation-rue* (*localization-street*) and *localisation-lieurelatif-general*, mistaken 10 and 6 times, respectively.

This simple qualitative analysis shows that our reranking models are really effective and can exploit complex information that baseline models, based on local information, cannot in general use.

B. Overall Contribution

In this paper, we have described several approaches to SLU, with particular emphasis on discriminative reranking, which exploits SLU hypotheses from baseline models, e.g. SFST and CRF. The main characteristics of our methods are: first, we approached SLU as a semantic parsing reranking, which is different from syntactic parsing reranking. Thus, we designed and studied different kernels on structures that are not syntactic. Indeed, we use semantic structures, which aim at representing lexical semantics and the relationships between semantic components (i.e. concepts).

Second, we automatically construct our structures on noisy data, which, in contrast with typical dependency or constituency syntactic structures, are designed to be robust to noise.

Third, we designed and tested new kernels for semantic tree processing, e.g. the kernels resulting from the application of PTK to our new designed conceptual tree structures (which result in different kernels from those in [31]). These, as shown by our experiments, are much more effective than other kernels. We also experimented with string kernels to provide another non-hierarchical semantic model, whose low efficiency motivated the structuring of concept semantics in trees rather than in sequences. In other words, our hierarchical semantic definition is a step towards the design of compositional semantics in noisy data.

Finally, the kind of features encoded by our kernels are n-grams of any size also containing gaps, which allow for including all possible long distance dependencies in the model, e.g. the relation between two departure cities. Such features, implicitly generated by our kernels, describe global semantics of the sentences annotated by baseline SLU models, therefore enabling global inference. The advantage of using kernels is that we do not need to manually analyze the data and intuitively choose the features that we believe may be effective.

C. Related Work

Among learning algorithms, CRF are one of the most useful method to take into account many features and their dependencies. However, in standard CRF or other non-kernel based approaches it is difficult to include interesting long-distance dependency features (or just effective n-grams) since either we have to manually define them (and this is a difficult task) or we have to include all possible n-grams. The latter choice makes learning impractical (too many features). Therefore, most implementations of CRF [40], [42], [43] use features in a limited window around the current word to be labeled (to limit the overall amount of memory and processing time).

Additionally, the CRF computational complexity when using features built on top of labels (concepts in our case) exponentially depends on the number of labels used to design such features. This limits the use of dependencies between features and labels (in most implementations at most bigram features are used) so that only approximated models are available, e.g. the skip-chain CRF [44].

One solution to solve the above limitations is the use of feature selection. Given the huge number of features involved in current sequence labeling tasks, wrapper approaches [45] are not viable (see for example [36]) thus only filter or embedded methods were studied, e.g. [46], [47]. Some interesting approaches to dependency feature extraction were proposed, e.g. in [3]–[5] and [6]. Finally, feature selection was also implemented within CRF using $l1$ regularization [48], [49], or laplacian prior [43], [50]. These methods allow for effective feature selection from a huge space, making learning with CRF feasible even with billions of features. Unfortunately, including higher order label features, such as concept dependencies, is still problematic. To our knowledge the only remarkable work in this direction is described in [51].

Other relevant work related to our article concerns with reranking. In [31], tree kernels for reranking of syntactic parse trees were applied whereas in [52] subtrees were efficiently used in an explicit structure space. Hidden-variable models were studied in [53], where a significant improvement was reached by exploiting several manually designed features. This is not always possible for new tasks/domains, like ours. Our approach, as we previously pointed out, is completely different (with respect to tree type and kernels). From a conceptual point of view, the work on [54], [55] is more similar to ours as it models the extraction of semantics as reranking problem also using string kernels. However, for such purpose the manual annotation of Minimal Representation Semantic trees (which are expensive to produce) is needed. Moreover, the studied application domain, coaching instructions in robotic soccer and a natural language database interface, is very limited compared to ours.

A more similar task has been studied in [56] for boundary detection in conversational speech. The significant improvement over the baseline model shows that reranking is an interesting approach for SLU. In our paper, for the first time, we provide indications on how designing DRMs by exploiting the potential of kernel methods. This has been carried out by also capitalizing our experience in other researches, e.g. [17], [19] and [57].

Previous work shows that reranking is an effective framework to improve and encode global knowledge about the target task. One of its drawback is the upperbound on system accuracy set by the baseline model hypotheses. In other words, if the base model does not provide optimal hypotheses in the top n positions, no reranker can achieve large improvement. Thus it is important to analyze such upperbound given by the error rate computed when always the best hypothesis is selected (by an oracle). We report the Oracle Error Rates in Table XIV for the Italian LUNA and for the French MEDIA corpora. Its entries clearly show that the upperbounds defined by the base-model hypotheses are largely above the state-of-the-art, i.e. there is still a wide margin for improvement.

Note that the state-of-the-art SLU on MEDIA combines several hypotheses with a ROVER approach [58] at token level. Although this should provide more general and flexible solutions, the oracle accuracy of our reranking at sentence level is far much higher. Thus before the use of a reranker at sentence level becomes a limitation, we have to prove that the other methods can at least remotely approach such oracle accuracy. The high value of the latter is partially due to the application of our simple but effective search in the hypothesis space, i.e. the SIM. Finally, in such perspective, other approaches considering the whole hypothesis set have been studied, e.g. [59].

Regarding our previous work, we designed some preliminary reranking models based on kernel methods in [11]–[13] although the used hypotheses were only generated by FSTs. In this paper, we have firstly proposed (i) CRF-based rerankers, (ii) a much more extensive experimentation on also new corpora, i.e. ATIS and an extended version of LUNA, and (iii) the new valuable approaches, i.e. RRS and SIM.

D. Final Remarks

In this section, we summarize the outcome of our comparative analysis, carried out on three different tasks: ATIS, MEDIA and LUNA:

- Our best DRMs consistently and significantly improve the respective baseline model (SFST or CRF) on all corpora where our CRF baseline model used the same setting and obtained the same accuracy of the state-of-the-art (reported in previous work).
- Experiments with automatic speech transcriptions revealed the robustness of the reranking models to transcription errors.
- The reranking model, using kernels for NLP like String and Tree Kernels can take into account arbitrarily long distance dependencies of words and concepts.
- Kernel methods show that combinations of feature vectors, sequence kernels and other structural kernels, e.g. on shallow or deep syntactic parse trees, appear to be a promising research line.

Our DRMs reach high accuracy thanks to two interesting strategies we propose for improving SLU reranking:

- The hypothesis selection criteria, i.e. SIM. This allows for selecting hypotheses from a large set, i.e. those that are most likely to be correct.
- The ReRank Selection strategy, which is based on the scores of the baseline and reranking models. This allows for recovering from mistakes of the reranking models, i.e. in case the top ranked hypothesis after reranking is less correct than the original best hypothesis.

In the future this research could be extended by focusing on advanced shallow semantic approaches such as predicate argument structures, e.g. [19]. Additionally, term similarity kernels, will be likely improve reranking models, especially when combined with syntactic tree kernels, e.g. [60]. Another interesting future work would be the use of more than one model to generate hypotheses for learning the reranker so that several approaches can be combined similarly to ROVER methods (like in [40]).

Finally, given the latest results on reverse-kernel engineering [35], it would be possible to extract meaningful features from our reranking models and use them in other state-of-the-art approaches, e.g. CRF. At the same time, methods to use kernels in CRF have been developing [61]. The reverse engineering will also allow for obtaining faster approaches. Alternative methods to design fast training may follow the research line in [33]. On the dialog perspective, our improved SLU system could be combined with [62] for the design of an effective dialog manager.

REFERENCES

[1] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding: A survey," *IEEE Signal Processing Magazine*, pp. 50–58, 2008.

[2] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, 2001, pp. 282–289.

[3] M. Jeong and G. G. Lee, "Multi-domain spoken language understanding with transfer learning," *Speech Communication*, pp. 412–424, 2009.

[4] ——, "Exploiting non-local features for spoken language understanding," in *Proceedings of COLING*, 2006, pp. 412–419.

[5] H. Chieu and H. Ng, "Named entity recognition: a maximum entropy approach using global features," in *Proceedings of COLING*, 2002, pp. 190–196.

[6] J. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of ACL*, 2005, pp. 363–370.

[7] C. Raymond, F. Béchet, R. De Mori, and G. Damnati, "On the use of finite state transducers for semantic interpretation," *Speech Communication*, pp. 288–304, 2006.

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

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

[10] M. Dinarelli, S. Quarteroni, S. Tonelli, A. Moschitti, and G. Riccardi, "Annotating spoken dialogs: from speech segments to dialog acts and frame semantics," in *Proceedings of SRSL Workshop of EACL*, 2009.

[11] M. Dinarelli, A. Moschitti, and G. Riccardi, "Re-ranking models based on small training data for spoken language understanding," in *Proc. of EMNLP*, 2009, pp. 11–18.

[12] ——, "Re-ranking models for spoken language understanding," in *Proc. of EACL*, 2009, pp. 202–210.

[13] ——, "Concept segmentation and labeling for conversational speech," in *Proc. of Interspeech*, 2009.

[14] A. Moschitti, "A study on convolution kernels for shallow semantic parsing," in *Proceedings of ACL'04*, 2004.

[15] ——, "Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees," in *Proceedings of ECML*, 2006, pp. 318–329.

[16] ——, "Making tree kernels practical for natural language learning," in *EACL 2006*, 2006.

[17] A. Moschitti, D. Pighin, and R. Basili, "Semantic role labeling via tree kernel joint inference," in *Proc. of CoNLL*, 2006.

[18] A. Moschitti, "Kernel methods, syntax and semantics for relational text categorization," in *Proceeding of CIKM*, 2008.

[19] A. Moschitti, D. Pighin, and R. Basili, "Tree kernels for semantic role labeling," *Computational Linguistics*, pp. 193–224, 2008.

[20] A. Moschitti, "Syntactic and semantic kernels for short text pair categorization," in *Proceedings of EACL 2009*, 2009.
[21] D. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the scope of the atis task: the atis-3 corpus," in *Proceedings of HLT*, 1994, pp. 43–48.
[22] H. Bonneau-Maynard, C. Ayache, F. Bechet, A. Denis, A. Kuhn, F. Lefèvre, D. Mostefa, M. Qugnard, S. Rosset, and J. Servan, S. Vilaneau, "Results of the french evalda-media evaluation campaign for literal understanding," in *LREC*, 2006, pp. 2054–2059.
[23] S. Quarteroni, G. Riccardi, and M. Dinarelli, "What's in an ontology for spoken language understanding," in *Proc. of Interspeech*, 2009.
[24] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proc. of Interspeech*, 2007, pp. 1605–1608.
[25] Y. He and S. Young, "Semantic processing using the hidden vector state model," *Computer Speech and Language*, pp. 85–106, 2005.
[26] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Proceedings of the VCL Workshop*, 1995, pp. 84–94.
[27] M. Jeong and G. G. Lee, "Practical use of non-local features for statistical spoken language understanding," *Comput. Speech Lang.*, pp. 148–170, 2008.
[28] I. Tsochantaridis, T. Hofmann, T. Joachims, , and Y. Altun, "Support vector learning for interdependent and structured output spaces," in *Proc. of ICML*, 2004.
[29] L. Shen and A. Joshi, "An SVM based voting algorithm with application to parse reranking," in *In Proc. of CoNLL*, 2003, pp. 9–16.
[30] R. Herbrich, T. Graepel, and K. Obermayer, *Large Margin Rank Boundaries for Ordinal Regression*. MIT Press, Cambridge, MA, 2000.
[31] M. Collins and N. Duffy, "New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete structures, and the voted perceptron," in *Proceedings of ACL*, 2002, pp. 263–270.
[32] S. Libin, A. Sarkar, and F. Och, "Discriminative reranking for machine translation," in *HLT-NAACL*, 2004, pp. 177–184.
[33] A. Severyn and A. Moschitti, "Large-scale support vector learning with structural kernels," in *ECML*, 2010, pp. 229–244.
[34] ——, "Fast support vector machines for structural kernels," in *ECML*, 2011.
[35] D. Pighin and A. Moschitti, "Reverse engineering of tree kernel feature spaces," in *Proc. of EMNLP*, 2009.
[36] M. Collins and T. Koo, "Discriminative re-ranking for natural language parsing," *Computational Linguistic (CL)*, pp. 25–70, 2005.
[37] A. Stolcke, "Srilm: an extensible language modeling toolkit," in *Proceedings of SLP*, 2002.
[38] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Technical Report of Computer Science Group*, 1998.
[39] T. Kudo and Y. Matsumoto, "Chunking with support vector machines," in *Proc. of NAACL*, 2001, pp. 1–8.
[40] S. Hahn, P. Lehnen, and H. Ney, "System combination for spoken language understanding," in *Proc. of Interspeech*, 2008, pp. 236–239.
[41] A. Yeh and K. Church, "More accurate tests for the statistical significance of result differences," 2000.
[42] T. Kudo, "CRF++ toolkit," 2005, http://crfpp.sourceforge.net/.
[43] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proc. of ACL*, 2010, pp. 504–513.
[44] C. Sutton and A. McCallum, "Collective segmentation and labeling of distant entities in information extraction," in *In ICML workshop on Statistical Relational Learning*, 2004.
[45] R. Kohavi and G. John, "Wrappers for feature subset selection," *ARTIFICIAL INTELLIGENCE*, pp. 273–324, 1997.
[46] A. McCallum, "Efficiently inducing features of conditional random fields," in *Proc. of Uncertainty in AI*, 2003.
[47] R. Klinger and C. Friedrich, "Feature subset selection in conditional random fields for named entity recognition," in *Proc. of RANLP*, 2009, pp. 185–191.
[48] J. Gao, G. Andrew, M. Johnson, and K. Toutanova, "A comparative study of parameter estimation methods for statistical natural language processing," in *Proc. of ACL*, 2007, pp. 824–831.
[49] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proc. of ACL IJCNLP*, 2009, pp. 477–485.
[50] S. Riezler and A. Vasserman, "Incremental feature selection and l1 regularization for relaxed maximum-entropy modeling," in *EMNLP*, 2010.
[51] X. Qian, X. Jiang, Q. Zhang, X. Huang, and L. Wu, "Sparse higher order conditional random fields for improved sequence labeling," in *Proceedings of ICML*, 2009, pp. 849–856.
[52] T. Kudo, J. Suzuki, and H. Isozaki, "Boosting-based parse reranking with subtree features," in *ACL*, 2005.
[53] T. Koo and M. Collins, "Hidden-variable models for discriminative reranking," in *Proc. of EMNLP*, 2005, pp. 507–514.
[54] R. Ge and R. Mooney, "Discriminative reranking for semantic parsing," in *Proc. of COLING*, 2006, pp. 263–270.
[55] R. Kate and R. Mooney, "Using string-kernels for learning semantic parsers," in *Proc. of IJCNLP*, 2006, pp. 913–920.
[56] B. Roark, Y. Liu, M. Harper, R. Stewart, M. Lease, M. Snover, I. Shafran, B. Dorr, J. Hale, A. Krasnyanskaya, and L. Yung, "Reranking for sentence boundary detection in conversational speech," in *ICASSP*, 2006, pp. 545–548.
[57] A. Moschitti, D. Pighin, and R. Basili, "Tree kernel engineering for proposition reranking," in *Mining and Learning with Graphs (MLG)*, 2006.
[58] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER)," in *Proc. of ASRU*, 1997, pp. 347–352.
[59] L. Huang, "Forest reranking: Discriminative parsing with non-local features," in *Proc. of ACL*, 2008, pp. 586–594.
[60] S. Bloehdorn and A. Moschitti, "Structure and semantics for expressive text kernels," in *Proc. of CIKM*, 2007, pp. 861–864.
[61] J. Lafferty, X. Zhu, and Y. Liu, "Kernel conditional random fields: Representation and clique selection," in *in ICML*, 2004.
[62] L. Zettlemoyer and M. Collins, "Learning context-dependent mappings from sentences to logical form," in *Proc. of ACL and IJNLP*, 2009, pp. 976–984.

Marco Dinarelli received the Ph.D. in Information and Communication Technology in March 2010 from the International Doctoral School of University of Trento. The main topic of his Ph.D. thesis was Spoken Language Understanding for Spoken Dialog Systems, with particular focus on models integration via discriminative reranking. Marco Dinarelli is currently a post-doctoral research associate at LIMSI-CNRS (France), where he is working on NLP tasks for French.

Alessandro Moschitti is an assistant professor at Computer Science Department of the University of Trento. He took his PhD in Computer Science at University of Rome "Tor Vergata" in 2003, and he worked for two years as an associate researcher at the University of Texas at Dallas. His expertise concerns machine learning approaches to Natural Language Processing, Information Retrieval and Data Mining. He has devised innovative kernels within Support Vector and other kernel-based machines for advanced syntactic/semantic processing documented by more than 130 articles. These have been published in the major conferences and journals of Computational Linguistics, Machine Learning, Information Retrieval and Data Mining, for which he is also an active PC member/area chair. He has received several best paper awards and the IBM Faculty award.

Giuseppe Riccardi, Fellow, IEEE (M'96-SM'04-F'10) joined AT&T Bell Laboratories (1993) and AT&T Labs-Research (1996) where he worked in the Speech and Language Processing Lab. In 2005 he joined the University of Trento (Italy) where he is affiliated with the EECS Department and Center for Mind/Brain Sciences. He is the founder and director of the Signals and Interactive Systems Lab. He has co-authored more than 120 papers and 50 patents. His current research interests are language modeling and understanding, spoken/multimodal dialog, affective interfaces, machine learning and machine translation. He has been on the scientific committee of the major speech and language processing conferences and journals. He received the Marie Curie Research Excellence grant by the European Commission and the IEEE SPS Best Paper Award.