Global Machine Learning for Spatial Ontology Population

Parisa Kordjamshidi, Marie-Francine Moens

KU Leuven, Belgium

Abstract

Understanding spatial language is important in many applications such as geographical information systems, human computer interaction or text-to-scene conversion. Due to the challenges of designing spatial ontologies, the extraction of spatial information from natural language still has to be placed in a well-defined framework. In this work, we propose an ontology which bridges between cognitive-linguistic spatial concepts in natural language and multiple qualitative spatial representation and reasoning models. To make a mapping between natural language and the spatial ontology, we propose a novel global machine learning framework for ontology population. In this framework we consider relational features and background knowledge which originates from both ontological relationships between the concepts and the structure of the spatial language. The advantage of the proposed global learning model is the scalability of the inference, and the flexibility for automatically describing text with arbitrary semantic labels that form a structured ontological representation of its content. The machine learning framework is evaluated with SemEval-2012 and SemEval-2013 data from the spatial role labeling task.

Keywords:
Spatial information extraction, Text mining, Structured output learning, Ontology population, Natural language processing

1. Introduction

An essential function of natural language is to talk about the location and translocation of objects in space. Understanding spatial language is important in many applications such as geographical information systems (GIS), human computer interaction, text-to-scene conversion, and representation and extraction of spatial information from web resources such as travelers blogs or websites about tourism. Due to the complexity of spatial primitives and notions, and the challenges of designing ontologies for formal spatial representation, the extraction of the spatial semantics from natural language still has to be placed in a well-defined framework.

We have two main contributions toward solving this problem. The first contribution is that we propose a spatial ontology based on two layers of semantics. This ontology is based on a previously proposed spatial annotation scheme by the authors [1]. Its first layer is based on commonly accepted cognitive spatial notions and the second is based on multiple well-known qualitative spatial reasoning models. An automatic mapping to such an ontology bridges between natural language and qualitative spatial representation and reasoning models, which makes automatic spatial reasoning based on spatial information in linguistic expressions feasible. This ontology can be integrated in larger ontologies, for example, to represent spatial meaning in unstructured data in the context of the Semantic Web.

The second contribution of this work is that we propose a novel global supervised machine learning model for spatial ontology population. For this supervised learning framework, we build rich annotated corpora and an evaluation scheme. We point to the linguistic features and structural characteristics of spatial language that aid the use of

machine learning. We view ontology population as a means for creating meaning representations from text. In this model the segments of the input text are described by semantic abstractions or concepts and their relationships defined by the ontology, which form the output space of the learning problem [2]. In the proposed global learning framework, the ontology components including spatial roles and their relations, and multiple formal semantic types are learned while taking into account the ontological constraints and the structural characteristics of the spatial language.

Learning a model that considers the global correlations between the output components usually becomes computationally complex. To deal with the complexity in training and prediction phases, we use an efficient inference approach based upon combinatorial optimization techniques for both phases. This approach can deal with a large number of variables and constraints, and makes building a structured machine learning model for ontology population, feasible.

We decompose the learning problem into simpler problems that are jointly optimized. We propose a technique which we call *communicative inference* based on the ideas of *alternating optimization* for solving smaller subproblems of the main objective function [3]. Each subproblem is solved by using linear programming (LP) solvers and the subproblems communicate to each other by passing the local solutions. We show that the suggested framework is beneficial compared to local learning as well as compared to pipelining the independently learned models for the concepts in the ontology. The proposed inference approach makes the global learning scalable.

The application of the global machine learning model for ontology population is not limited to the extraction of spatial semantics; it could be used to populate any ontology. Moreover, due to decomposing the ontology to its solvable parts, this approach is scalable to be applied for approximate global learning for large ontologies of the Semantic Web. We argue therefore that this work is an important step towards automatically describing text with semantic labels that form a structured ontological representation of the content.

Our extensive experimental study using the spatial ontology indicates the advantage of global learning while considering ontological constraints and structural characteristics of the spatial language compared to learning local models for the various parts of the ontology independently. The experiments are performed using the corpora provided by the SemEval-2012 and SemEval-2013 shared task on spatial role labeling.

In Section 2, we provide the problem definition and the spatial ontology population task in its two layers of semantics. In Section 3, we discuss the features and constraints that are useful for learning the spatial ontology population. A background to structured learning is provided in Section 4. The proposed structured learning model for spatial ontology population is described in Section 5. The proposed inference approach is explained in Section 6. Section 7 specifies the details of the components of the spatial ontology population model. The various designed local and global models are clarified in Section 8. Section 9 presents the experimental results. An overview of the related research is provided in Section 10. We draw conclusions, set our work in a broader context, and point to the future extensions in Section 11.

2. General Problem Definition

We define a framework for mapping natural language to spatial ontologies. Although pragmatic, our proposed framework is based on the theoretical cognitive and linguistic foundations, as well as on cognitively adequate formal spatial models. The task is formulated as an ontology population to be performed via supervised machine learning models. We aim at learning to assign the segments in the sentence to the concepts in the ontology. The considered concepts form a *light weight* ontology which is based on a previously proposed spatial annotation scheme by the authors [1]. We highlight the distinction between two *spatial role labeling* (SpRL) and *spatial qualitative labeling* (SpQL) layers in the ontology. We describe the structural characteristics of the two-layered ontology to be exploited in the learning models.

2.1. Two Layers of Semantics

Spatial language can convey complex spatial relations along with polysemy and ambiguity inherited in natural language. Linguistic spatial expressions can express various aspects of the space at the same time [4]. In contrast to natural language, formal spatial models focus on one particular spatial aspect such as orientation, topology or distance and specify its underlying spatial logic in detail [5, 6]. Therefore there is a gap between the level of expressivity and specification of natural language and spatial calculi models [7].

Due to this gap, learning how to map the spatial information in natural language onto a formal representation is a challenging problem. However, such a mapping is useful because formal spatial models enable automatic spatial

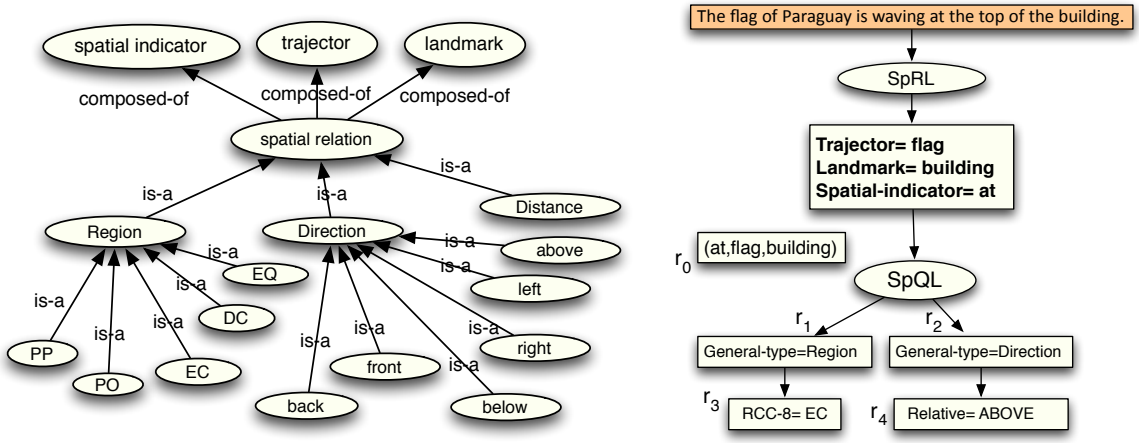


Figure 1. (a) The spatial ontology. (b) Example sentence and the recognized spatial concepts.

reasoning that is difficult to perform on natural language expressions. To overcome the complexity of this problem in a systematic way, our spatial ontology is divided into two abstraction layers [7, 8, 9]:

1. A layer of **linguistic conceptual representation** called spatial role labeling (SpRL), which predicts the existence of spatial information at the sentence level by identifying the words that play a particular spatial role as well as their spatial relationship [10];
2. A layer of **formal semantic representation** called spatial qualitative labeling (SpQL), in which the spatial relation is described with semantic attributes based on qualitative spatial representation models (QSR) [11, 12].

In our conceptual model we argue that mapping the language to multiple spatial representation models could solve the problem of the existing gap between the two layers to some extent (also see [13, 14] in the context of robotics and navigational instructions). Because various formal representations capture the semantics from different angles, their combination covers various aspects of spatial semantics needed for locating the objects in the physical space. Hence, the SpQL has to contain multiple calculi models with a practically acceptable level of generality. However, we believe that this two layered model does not yet yield sufficient flexibility for ideal spatial language understanding. As in any other semantic task in natural language, additional layers of *discourse* and *pragmatics* must be worked out, which is not the focus of this work.

2.2. Task Definition as Ontology Population

Our main task is to map a given sentence x composed of a number of words $x_1 \dots x_n$ to the predefined spatial ontology shown in Figure 1(a). The task is to label the words in the sentence with spatial roles (SpRL), detect the spatial relations, and label the spatial relations with their spatial semantics including course-grained as well as fine-grained semantic labels. The words can have multiple roles and the relations can have multiple semantic assignments. The labels are assigned according to the relationships and constraints that we discuss in the following sections. The considered spatial ontology here is only a lightweight [15] ontology, but pinpoints the main challenges in the recognition of ontological label structures in text.

2.2.1. Spatial Role Labeling (SpRL)

In the *spatial role labeling* (SpRL) layer the cognitive-linguistic spatial semantics based on the theory of *holistic spatial semantics* are considered [16, 17]. Figure 1(b) shows the sentence, *The flag of Paraguay is waving at the top of the building.*, which is labeled according to the nodes in the spatial ontology of Figure 1(a). In the SpRL step the goal is to identify the words that play a spatial role in the sentence and to classify their roles; moreover to recognize the link between the spatial roles and extract the spatial relations. In this sentence, we need to extract a spatial relationship signaled by *at* that holds between *flag* and *building*. The word *flag* has the role of *trajector* (*tr*). The trajector is an entity whose location is described. The word *building* has the role of *landmark* (*lm*). The landmark is a reference

object for describing the location of a trajector. These two spatial entities are related by the spatial expression *at* that is the *spatial indicator* (*sp*). The spatial indicator signals the existence of spatial information in the sentence.

These spatial roles are the three main nodes in our ontology. We refer to these nodes as *single label*s. A *single label* refers to an independent concept in the ontology. For a spatial configuration, we consider the link between the three roles, which is labeled as a *s*patial relation, also called a spatial triplet. We refer to these kind of nodes in the ontology as *linked label*s. *Linked label*s show the connection between the concepts in the ontology. For example the spatial relation is a linked label that shows a *composed-of* relationship with the composing labels of spatial roles. There is one spatial relation in the above sentence, <at_{sp} $flag_{tr}$ $building_{lm}$>. In general, there can be a number of spatial relations in each sentence. Although the spatial indicators are mostly prepositions, in general the *sense* of the prepositions depends on the *context*. The first preposition *of* in the example sentence states the possession of the flag, so <*of flag Paraguay*> is not a spatial relation.

The trajectors and landmarks can be implicit, meaning that there is no word in the sentence to represent them. In some linguistic spatial expressions, there is no need to express the spatial information based on any landmark [17]. In these cases we use the term *undefined* instead of the roles to keep the spatial relation representation consistent. For example, in the sentence *Come over here* where the trajector *you* is only implicitly present, the spatial relation is represented as <$over_{sp}$ $undefined_{tr}$ $here_{lm}$>. The other SpRL elements such as *motion*, *path* and *frame of reference* are not considered due to the focus on the static descriptions contained in our main experimental dataset. For more information about these additional concepts see [18, 19].

Spatial relations can be inferred by spatial reasoning too. For instance, in the example of *The book is on the table behind the wall.* The spatial relations <on_{sp} $book_{tr}$ $table_{lm}$> and <$behind_{sp}$ $table_{tr}$ $wall_{lm}$> are extracted directly from the sentence but the relation <*behind book wall*> can be inferred. Such inferred relations are not considered in this work because they make the semantic annotation of the data more difficult and less consistent, but can be obtained through reasoning over the extracted information.

2.2.2. Spatial Qualitative Labeling (SpQL)

In the *spatial qualitative labeling* (SpQL) layer, the goal is to map the entire spatial configuration that is extracted from the SpRL layer to a formal semantic representation. As we simplified the first layer by ignoring a number of concepts such as shape, size and motion, in this layer also a number of simplifications are considered for the sake of feasibility of the learning task given the available resources and data. Our representation of the spatial semantics is based on multiple qualitative spatial calculi [6, 4]. Figure 1(a) shows the semantics that are considered in this work. The three general types of regional (i.e. topological), directional and distal cover all coarse-grained aspects of space (ignoring shape and size) and qualitative spatial calculi are available for them. Henceforth, we map extracted cognitive linguistic elements to multiple qualitative representations including these three categories. The fine-grained semantics of these types are described below.

Region. The topological relationships are represented by the *general type* of *region* in our ontology. Models from specific region-based qualitative spatial calculi can be used for this type. We use Region Connection Calculus RCC-8 [20], which is heavily used in qualitative spatial representation and reasoning. RCC-8 provides 8 basic relations (see Fig. 2): disconnected $\mathsf{DC}(\mathsf{a},\mathsf{b})$, externally connected $\mathsf{EC}(\mathsf{a},\mathsf{b})$, partial overlap $\mathsf{PO}(\mathsf{a},\mathsf{b})$, equal $\mathsf{EQ}(\mathsf{a},\mathsf{b})$, tangential proper-part $\mathsf{TPP}(\mathsf{a},\mathsf{b})$, non-tangential proper-part $\mathsf{NTPP}(\mathsf{a},\mathsf{b})$, tangential proper-part inverse $\mathsf{TPPI}(\mathsf{a},\mathsf{b})$, and non-tangential proper-part inverse $\mathsf{NTPPI}(\mathsf{a},\mathsf{b})$ which describe mutually exclusive topological relationships between two (well-behaved) regions in space. However, when humans use language to describe objects, the inverse proper part relations occur less frequently (cf. [21]). This motivated us to combine all variations of proper part including {*TPP,NTPP,TPPI, NTPPI*} into one class {*PP*}. Hence, we used five categories for topological relations in our ontology.

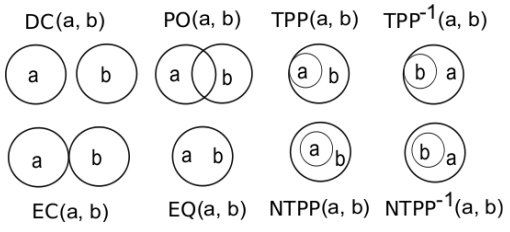


Figure 2. The RCC-8 relations.

Direction. For *direction*, according to the spatial scheme in [1], the *specific type* is assigned by a value in {*Absolute, Relative*}. The absolute directions are the geographical ones such as South, West and so on. The relative directions are {*Left, Right, Front, Behind, Above, Below*} which are used in qualitative direction calculus. In this work due to

the type of texts used in the experiments (i.e., image descriptions) absolute relations do not occur. Hence we only consider learning *relative* directional relations.

Distance. Distances are specified with {*Qualitative, Quantitative*} in the scheme of [1]. Our corpus contains a few examples with distal information. Therefore one general class of *distance* is used in our spatial ontology.

We assume that trajector and landmark are 'interpreted' as two regions. As a consequence, we can map static as well as dynamic spatial expressions, although dynamicity is not directly covered in RCC (unless neighborhood graphs are used as a form of spatial change over time [21]). Since the mapping can point to relations from different calculi (e.g. RCC and an orientation calculus), this is more suited to achieve the required level of expressivity for a spatial expression. Given multiple representations over the linguistic spatial information, qualitative (even probabilistic) spatial reasoning will be feasible over the produced output.

In the example of Figure 1(b) spatial roles compose the relation (at,flag,building) and then this relation is labeled with *region* and *EC* together with *direction* and *above*. No distal information is annotated for this sentence.

3. Constraints and Features for the Machine Learning Models

As in other computational linguistic tasks, the lexical, syntactic and semantic features of language can help with the extraction of spatial semantics. There is also linguistic and commonsense background knowledge on the spatial language to be exploited when designing an intelligent model for automatic spatial semantic extraction. In this section we aim to specify all types of information that can be useful for the machine learning models that we design. We divide these characteristics in two categories of *Features* and *Constraints*. The benefit of the distinctions between these two types of information is extensively discussed in [22]. Generally, constraints capture global/structural characteristics of the problem while the features capture local properties. Considering global properties when training an extraction model is more complex than considering only local properties. The complexity is due to the involvement of complex correlations between variables. Moreover, learning models need a large number of training examples in order to capture the global properties of the problem. Global characteristics also can be modeled in the form of features, but separating them from the features helps the models to treat them more efficiently [22].

3.1. Constraints

The spatial language has a number of structural characteristics that we exploit for automatic extraction of spatial roles and relations. These global constraints should hold among the predicted *output labels* assigned by the machine learning model. For instance:

(a) Each spatial relation is composed of three spatial roles where some of them can be undefined. It implies the following constraints that are referred to as *composed-of constraints*.

- If there is a trajector or landmark in the sentence then there is a spatial indicator too.
- If a spatial indicator is detected in the sentence then we impose the extraction of a spatial relation, possibly with undefined roles.

For example, if the learning model classifies the *building* as a trajector/landmark then it needs to find a relevant spatial indicator in the same sentence too. Similarly, if the model recognizes *at* as a spatial indicator, then it must find a relevant trajector and landmark in the sentence. If it does not find any word with enough positive features for playing a spatial role, it should predict a relation with *at* as indicator along with undefined trajector/landmark, or might classify *at* as not being a spatial-indicator.

(b) An object can play only one role of trajector or landmark with respect to one specific spatial indicator. We refer to this property as the *multilabel constraint*. This is a common sense property of the spatial language. For example, if we describe the location of the *book* using the preposition *on*, then it has the role of trajector and naturally can not be the landmark of the same preposition.

(c) Because this scheme does not include spatial reasoning the following structural property is imposed, also referred to as *spatial reasoning constraint*: each word with one type of spatial role is only connected to one spatial indicator. Note that a word can be connected to various spatial indicators while having different types of spatial roles with respect to those indicators. For example, in the sentence of Section 2.2.1, if *book* is linked to *table* as the trajector of *on*, then it can not be connected to *behind* as its trajector because such multiple connections with

a same role (trajector here) can happen only by spatial reasoning which is avoided in our annotations. However, *table* is connected to both indicators as it has the role of landmark with respect to *on* and the different role of trajector with respect to *behind*.

(d) The number of spatial relations in a sentence can be restricted. This restriction can be set according to the statistics in the annotated data and proportional to the length of the sentence. We refer to this type of constraint as the *counting constraint*. This is not a strict constraint and obviously depends on prior knowledge or the statistics that we might collect from our domain to help the learning models in their predictions just by reducing the number of possible predictions.

The above mentioned constraints are about the SpRL layer of the ontology and the sentence level characteristics of the spatial language. We refer to the SpRL layer constraints as *horizontal constraints*. Considering the ontological structure of the nodes in the SpQL layer, we introduce the following constraints:

(i) If an *is-a* relationship holds between two semantic nodes (labels) l_1 and l_2 in the ontology, that is l_1 is-a l_2, then each instance of l_1 should be an instance of l_2. We refer to this type of constraint as the *is-a constraint*. For example, according to the ontology of Figure 1, a learning model is not allowed to classify the spatial relation as *EC* without classifying it as *regional* at the same time.

(ii) Each arbitrary relation can be assigned zero or more fine-grained semantic labels. If the relation with three arguments is recognized as spatial, then there should be at least one fine-grained semantic label assigned to it. We refer to this type of constraint as the *null-assignment constraint*. For example, if $< at\ flag\ building >$ is recognized as a spatial relation, then the model is forced to find at least one best assignment for the spatial semantics of this relation.

(iii) Given a general type-label, each spatial relation is associated with only one fine-grained label under that type. This property is due to the mutual exclusivity of the formal spatial semantics. We refer to this type of constraint as the *mutual-exclusivity constraint*. For example, the relation between $< at\ flag\ building >$ can not be *EC* and *DC* at the same time given the mutual exclusivity property of the RCC. However, we allow multiple assignments at the upper level of the ontology. For instance, the above mentioned relation can have regional and directional semantics at the same time.

These last three types of constraints relate to the structure of the in-depth and fine-grained spatial semantics and properties of the formal models, hence we refer to these as *vertical constraints.*

3.2. Linguistic Features

In addition to the global characteristics of spatial language there are a number of linguistically motivated features used in training the learning models. We use natural language processing tools to process the sentences and extract these features. These *input features* describe single words or even composed components of a sentence, like pairs of words or phrases. We assume that words are indexed based on their position in the sentence. We call the features that are assigned to a single word, *local features*. Although for some local features the context of a word in the sentence has to be considered, we use the term local when the features are assigned as a property of one single word. The following local features are assigned to each single input component (word). We refer to the input features by the ϕ symbol, and each class of features is indexed by a relevant name, $\phi_{local}(x_i)$.

- **Word-form** $\phi_{wf}(x_i)$: Lexical information is indicative and important for spatial semantics. So the word form itself is used as a feature. For example, the two expressions, *the meeting in the afternoon* and *the meeting in the office*, differ only in one word, but the term *afternoon* can not be a landmark, while the term *office* can be.

- **Part-of-speech tag** $\phi_{pos}(x_i)$: POS tags are informative features for recognizing spatial roles, e.g. the IN tag (i.e. preposition) gives a higher chance to a word for being a spatial indicator, and the NN tag (i.e. noun) gives a higher chance for being a trajector or landmark. Therefore, POS tags can be used as distinguishing features for training a model. For example, *He is on the left.* compared to *He left you.*, the two similar word forms *left* can be distinguished using their part-of-speech tags. *left* in the first sentence is a noun (NN) and has a spatial meaning, but in the second it is a verb (VB) and does not carry any physical locative information.

- **Semantic role** $\phi_{srl}(x_i)$: Because the models for recognizing the semantic roles of the constituents of a sentence are trained on large corpora, the roles can provide information for spatial role labeling. For example, a preposition which is labeled as locative (LOC) by a semantic role labeler has a high probability of being a spatial indicator.

- **Dependency relation** $\phi_{dprl}(x_i)$: The labels assigned to words by dependency parsers are useful for spatial role labeling. For example, the SBJ label, meaning that a word is the subject of the sentence, gives a higher chance to that word for being a trajector. Or the label of NMOD, meaning that a word is a noun modifier, often indicates that the word does not carry any spatial role. The relations represented by dependency trees could be directly exploited in finding spatial links as well. We exploit only the dependency labels for the sake of reduced complexity.

- **Subcategorization** $\phi_{sub}(x_i)$: Subcategorization shows the sister nodes of the parent of a word in the parse tree. For example, for the preposition *on* in the phrase *on your left*, this feature is IN-NP. This provides information about the syntactical context of each word. Given this feature the learning model will know, for instance, whether a specific preposition is part of a prepositional noun phrase or part of a verb phrase. Hence the learning model can learn the frequent contextual patterns of spatial prepositions.

- **Spatial context of the preposition** $\phi_{spc}(x_i)$: In spite of the verbs and nouns, spatial terms and prepositions are usually a closed lexical class of words in many languages. Hence a list of terms such as directions *left, right* and other spatial terms can be easily collected. Here, for each spatial indicator the existence of a spatial term in its neighborhood is used as a feature. This feature helps to recognize the spatial sense of the prepositions which are in a spatial context. Moreover, it helps to detect the undefined landmarks. Normally, undefined landmarks occur in the context of directional phrases such as *on the left*. In these cases the landmark is implicit and depends on the spatial frame of reference. However if a directional phrase is followed by *of* then it means the phrase attributes an explicit landmark, for example, in the phrase *on the left of the room*. In other words, if the spatial context contains the second preposition *of* then the possibility of having an explicit landmark is higher. So we define a feature with three dimensions to distinguish whether a spatial indicator has a spatial context with a second preposition, whether it has a spatial context without a second preposition or no spatial context at all.

In addition to the local features, we use a number of relational pairwise features between the words, including,

- **before** $\phi_{before}(x_i, x_j)$: This feature indicates whether the position of a word which is a candidate for trajector (or landmark) is before a word which is a candidate for spatial indicator. Also it signals whether the position of the candidate trajector is before the candidate landmark.

- **distance** $\phi_{dis}(x_i, x_j)$: The relative distance between trajector (landmark) candidate x_i and the candidate spatial indicator x_j is defined as,

$$distance = \frac{\text{\#Nodes on the path between } x_i \text{ and } x_j}{\text{\#Nodes in the parse tree}},$$

  and the integer value of the inverted distance is used as a nominal feature.

All the above mentioned features, which are mostly nominal, are turned into binary vectors. If there is no preprocessing and candidate selection phase for the roles, the relational features should be computed for all possible pairs of words in the sentence. We find it useful to make a distinction between *relational features* and *contextual features*. A *contextual feature* is a feature which, in spite of not being local, is assigned to a single component without mentioning the other related component(s). For example the above mentioned *spatial context* is such a feature since it is a binary feature that indicates whether a spatial term exists in the neighborhood of a preposition, but there is no explicit relation referring to the identifiers of its neighborhood.

4. Structured Learning Setting

In learning models for structured output prediction, given a set of N input-output pairs of training examples $E = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} : i = 1..N\}$, we learn an objective function $g(x, y; W)$ which is a linear *discriminant* function

Algorithm 1 Cutting-plane for SVM-struct

1: Given training data: $E = (x^i, y^i)_{i=1}^N$; $Cost$, ϵ
2: $\mathcal{S}_i \leftarrow \emptyset \ \ \forall i = 1, \ldots, N$
3: **repeat**
4: **for** $i = 1$ to N **do**
5: $H(y) \triangleq \Delta(y^i, y) + W^T f(x^i, y) - W^T f(x^i, y^i)$
6: compute $\hat{y} = \arg\max_{y \in \mathcal{Y}} H(y)$
7: compute $\xi_i = \max\{0, \max_{y \in S_i} H(y)\}$
8: **if** $H(\hat{y}) > \xi_i + \epsilon$ **then**
9: $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{\hat{y}\}$
10: $W \leftarrow$ optimize primal over $\cup_i \mathcal{S}_i$
11: **until** no $\mathcal{S}_i$ has changed during iteration

defined over the combined feature representation of the inputs and outputs denoted by $f(x, y)$ [23]:

$$g(x, y; W) = \langle W, f(x, y) \rangle. \tag{1}$$

W denotes a weight vector and $\langle , \rangle$ denotes a dot product between two vectors. A popular discriminative training approach is to minimize the following convex upper bound of the loss function over the training data:

$$l(W) = \sum_{i=1}^{N} \max_{y \in \mathcal{Y}} (g(x^i, y; W) - g(x^i, y^i; W) + \Delta(y^i, y)), \tag{2}$$

the inner maximization is called loss-augmented inference and finds the so called most violated constraints/outputs (y) per training example. All discriminative structured prediction algorithms such as structured perceptron [24, 25], max-margin Markov networks [26] and structured SVMs [23] need a solution for this crucial inference task during training to minimize $l(W)$. We apply structured support vector machines (SSVM), structured perceptrons (SPerc) and averaged structured perceptrons (AvGSPerc). We show the following 1-slack margin rescaling version of the **structured support vector machine** in which the margin is rescaled by the loss [23],

$$\begin{array}{ll} \min_{W,\xi} & \frac{1}{2}\|W\|^2 + \frac{Cost}{N} \sum_{i=1}^{N} \xi_i \\ s.t. & \forall i : \xi_i \geq 0, \\ \forall i, \forall y \in \mathcal{Y} : & \langle W, f(x^i, y^i) - f(x^i, y) \rangle \geq \Delta(y^i, y) - \xi_i, \end{array} \tag{3}$$

where ξ_is are the slack variables to allow errors in the training set particularly when the training data is not linearly separable and $Cost > 0$ is a constant or regulation parameter that controls the tradeoff between the training error minimization and margin maximization. Algorithm 1 shows the cutting plane algorithm of SVM-struct[1]. This algorithm suggested in [23] takes the most violated ys by finding the maximum a posteriori (MAP) of H in line 6. H is the objective of the loss-augmented inference defined in line 5. The algorithm adds the most violated examples one by one, building a working subset of constraints ($\cup_i \mathcal{S}_i$, $\mathcal{S}_i$ is the constraint set associated with the ith training example) at each iteration, and updates the weight vector W (line 10) by working on the primal formulation in this algorithm.

The **structured perceptron** has a very similar basis compared to the SSVM [25] and is shown in Algorithm 2. This algorithm minimizes the same convex upper bound $l(W)$ of the structured loss. In the simplest case there is no regularization term included and the training is performed by a sub-gradient algorithm. Beginning with a weight vector W initialized with zeros, the structured perceptron algorithm iterates through each element of the training set, updating the weight vector after processing each training instance. The training set is processed repeatedly until convergence. In each update step t, if the most violated y is not the correct answer, the difference between the feature vectors of the ground-truth and the model's prediction is added to the weight vector W.

In the **averaged structured perceptron** [24, 25] the final weight vector W is the average over all model weights W_t at each iteration t, that is $W = 1/\mathcal{T} \sum_{t=1}^{\mathcal{T}} W_t$, where $\mathcal{T}$ is the number of iterations. This heuristic regularizes the parameters of the model and compensates for not having an explicit regularization term in the training objective.

[1] http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

Algorithm 2 Sub-gradient-descent for structured perceptron

Given training data: $E = (x^i, y^i)_{i=1}^N$; step sizes η_t
$W \leftarrow \mathbf{0}$
for $t = 1$ to $\mathcal{T}$ **do**
  for $i = 1$ to N **do**
    $\hat{y} \leftarrow \arg\max_{y \in \mathcal{Y}} \Delta(y^i, y) + W^T f(x^i, y) - W^T f(x^i, y^i)$
    $W \leftarrow W + \eta_t (f(x^i, y^i) - f(x^i, \hat{y}))$

5. Link-And-Label Model

We aim to provide a simple and useful abstraction for designing global structured learning models for ontology population from text that is easily integrated in the above non-probabilistic structured output prediction models. We specify the learning components including input, output, joint feature function, global constraints, loss and inference in a framework which we name *Link-And-Label* (LAL) framework. The Link-And-Label name is inspired by the conceptualization process that a human does when extracting information from text and which is also the main function of ontologies. We usually group objects which are linked to each other with certain types of relationships and we label them as a more abstract concept. In our case the various segments of the text are linked to each other and labeled as an instance or an indicator of a specific concept. The labels themselves are the new properties of the higher level concepts, therefore by linking a number of labels (for example, in the case of a composed-of relationship) we build more complex concepts and new labels (such as a spatial relation).

Concepts can have various relationships over which the ontologies are designed. The relationships between concepts describe the relationships between the instances of them. This feature stimulates treating ontology population as a relational learning problem exploiting the ontology as a first order representation of the output space. To explain the Link-And-Label model, first we describe the terminology that we use based on the *input* and *output* distinction. Then we describe the objective function of the training and the prediction for ontology population in this framework.

5.1. Input Space

Each input x is a set of components $\{x_1..x_K\}$. Each component has a *type*. Each $x_k \in x$ is described by a vector of features relevant for its type. The feature vector is denoted by ϕ_p where p is an index that refers to a specific type. For instance, in semantic labeling of text an input type can be a word (atomic component) or a pair of words (composed component). Each type is described by its features (e.g. a single word by its part-of-speech, the pair by the distance of the two words). As described above, the features that represent a property of an atomic component are called local and the ones that describe the relation between more than one atomic component are called relational features.

5.2. Output Space

The output space y is represented by a set of *labels* $\mathbf{l} = \{l_1, .., l_P\}$. The labels are defined based on the nodes in the ontology and have ontological relationships to each other. To be able to represent complex output concepts in general for any arbitrary task, we distinguish between two types of labels, the *single labels* and *linked labels* that refer to an independent concept and to a configuration of a number of related single labels respectively. Linked labels can represent different types of semantic relationships between single labels. They can express *composed-of*, *is-a* and other semantics given in the ontology. For convenience, to show which labels are connected by a *linked label*, we represent the *linked labels* by a *label string*. This is a concatenation of the labels that are linked together and construct a bigger semantic part of the whole output. For example in Figure 1, a *spatial relation* can be denoted by *sp.tr.lm* meaning that it is *composed of* the three single labels, *sp*, *tr* and *lm*. Label strings can imply *is-a* or other semantic relationships between labels.

Definition. A label l is a sub-label of a label l' when all single labels that occur in the label string of l also occur in the label string of l', denoted by $l \prec l'$. In this case we call l' a super-label of l.

For example in Figure 1, *sp* is a sub-label of *sp.tr.lm*.

5.3. Connecting Input and Output Spaces

In the LAL model, labels are binary indicators that receive an input component and indicate whether it has that certain label. The binary indicator function for each linked label is defined according to its semantics. For example, a spatial relation label is defined in a way to convey the *composed-of* semantics based on the labels of its components. We use both notations of $l_p(x_k)$ or shorter l_{pk} to indicate the membership of the component x_k in the set of components with label l_p. To formally specify the connections between input components and output labels we use the notion of *template* as in relational graphical models [27, 28, 29]. The learning model is specified with a set of templates $C = \{C_1, .., C_P\}$. Each template $C_p \in C$ is specified by three main characteristics,

- *A subset of joint features*. This is referred to as local joint feature function and defined over a number of input type(s) and output label(s) associated with the template C_p. It is denoted by $f_p(x_k, l_p)$, where x_k is an input single or composed component, and l_p is a single or linked label that is entailed from the ontology labels $\mathbf{l}$.
- *Candidate generator*. It generates candidate components upon which the specified subset of joint features is applicable. The set of candidates for each template is denoted as C_{l_p}.
- *A block of weights* W_p. This is a block of the main weight vector W of the model which is associated with that template and its local joint feature function.

LAL objective function. The main objective *discriminant function* $g = \langle W, f(x, y)\rangle$, is a linear function in terms of the combined feature representation associated with each candidate input component and an output label according to the template specifications. So it is written in terms of the instantiations of the templates and their related blocks of weights W_p in $W = [W_1, W_2, .., W_P]$,

$$g(x, y; W) = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, f_p(x_k, l_p)\rangle = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, \phi_p(x_k)\rangle l_{pk} = \sum_{l_p \in \mathbf{l}} \langle W_p, \sum_{x_k \in C_{l_p}} (\phi_p(x_k) l_{pk})\rangle \tag{4}$$

where the local joint feature vector $f_p(x_k, l_p)$, is an instantiation of the template C_{l_p} for candidate x_k. This feature vector is computed by scalar multiplication of the input feature vector of x_k (i.e. $\phi_p(x_k)$), and the output label l_{pk}. This output label is the indicator function of label l_p for component x_k. Each indicator function of a template linked label is applied on the relevant input component and its value is one when the intended semantics behind it holds for that component. For example, if a template is an *and template*, it means the indicator function of the linked label is one if all included single label indicators are one when applied on the input parts. In this case we can represent the linked labels with the scalar product of the indicators of the sub-labels when forming the objective g. We can view the inference task as a *combinatorial constrained optimization* given the polynomial g which is represented in terms of labels, subject to the constraints that describe the relationships between the labels. For example, the *composed-of* relation between a linked label l denoted by the *label string* $l = l_i..l_j$, and its single sub-labels can be represented by the following constraint,

$$(l(x_c) = 1) \Rightarrow (l_i(x_1) = 1) \wedge .. \wedge (l_j(x_n) = 1)$$

where each label $l_{\_}$ applies only on a relevant type of component $x_{\_}$ and $x_1, .., x_n \subseteq x_c$; or the *is-a* relationships can be defined as the following constraint,

$$(l(x_c) = 1) \Rightarrow (l'(x_c) = 1)$$

where l and l' are two distinct labels that are applicable on the components with the same type of x_c. These are two commonly used ontological relationships also used in our spatial ontology, but other ontological relationships can be represented and directly exploited in the learning model.

It should be noted that, in general for designing such models, highly correlated labels should ideally be linked to each other and be considered in one template. However, considering the global correlations in one template can become very complex. Hence, these global correlations can be modeled via adding constraints. The constraints can hold between the instantiations of one template which implies the relations between the components of one type, also referred to as *autocorrelations* as defined in a *relational dependency networks* [30]. The constraints are exploited during training in the loss-augmented inference and are imposed on the output structure during prediction. In practice, we treat the objective as a linear function in which the association between linked labels and single labels in addition to their global relationships are modeled via linear constraints. When we need to do inference over an input example, we build a new instance of the objective function and the constraints.

Algorithm 3 Communicative inference

Given a training/test example (x, y) and the objective function $H(y)$
Given a decomposition over y as $S^x = \{y_1, y_2\}$, $\{H(y) \triangleq H(y_1, y_2)\}$
Given two disjoint subsets of constraints c_1 and c_2 over y_1 and y_2
$t \leftarrow 0$
Initialize y_1^t, y_2^t
repeat
  $t \leftarrow t + 1$
  $y_2^t \leftarrow \arg\max_{y_2} H(y_1^{t-1}, y_2)$ {An LP-relaxation subproblem subject to c_2}
  $y_1^t \leftarrow \arg\max_{y_1} H(y_1, y_2^t)$ {An LP-relaxation subproblem subject to c_1}
until $(y_1^{t-1} = y_1^t \wedge y_2^t = y_2^{t-1}) \vee t > Tmax$
$\hat{y} \leftarrow [y_1^t, y_2^t]$ { $\hat{y}$ is the MAP of H over y}

5.4. Component Based Loss

We define the loss function (Δ) that is decomposable in the same way as the joint feature function. This is to avoid increasing the complexity of the *loss-augmented* inference compared to the prediction time inference [26]. We define a component-based loss for each template label l_p by measuring the Hamming loss between the vector of predicted labels for all candidates (denoted by Λ_{l_p}) and the ground truth assignments (denoted by Λ'_{l_p}) and normalize by the number of candidates,

$$\Delta_{l_p}(\Lambda, \Lambda') = \frac{1}{|C_{l_p}|} \sum_{k=1}^{|C_{l_p}|} \Delta_H(l_{pk}, l'_{pk}) \text{ where } \Delta_H(l_{pk}, l'_{pk}) = l_{pk} + l'_{pk} - 2 l_{pk} l'_{pk}. \tag{5}$$

In addition to the complexity issues, using this loss function is very natural for ontology population because we basically perform collective classification of input components with respect to the nodes in the ontology and we jointly minimize a simple 0/1 loss for all label assignments. If some labels have priority for an application, the Hamming loss can be simply weighted based on the priorities:

$$\Delta(y, y') = \sum_{p=1}^{P} \omega_{l_p} \Delta_{l_p}(\Lambda_{l_p}, \Lambda'_{l_p}), \tag{6}$$

where ω_{l_p} is the weight of each template label l_p and P is the number of templates. In this way, the loss is still expressed in terms of the labels, and the two inference problems for training and prediction are similar (in terms of variables and constraints). Providing efficient solutions for inference over these objectives is the subject of the next section.

6. Communicative Inference

Solving the LAL objective function, given in Equation 2, during training of the model can become highly inefficient for most relational data domains. This is because the objective function and the constraints are in fact expressed in a first order representation (i.e. templates and types), and the corresponding ontologies or output label structures often produce a large number of output labels and constraints when instantiated for each training example. To solve this problem we propose an additional layer of decomposition as a meta frame for applying off-the-shelf linear programming (LP) solvers. There are a number of general ideas and various techniques for decomposing large structured learning problems that are not solvable with one standard technique due to computational complexity [31].

We propose an approach for decomposing the prediction and training time inference, which we call *communicative inference*. The basic idea is that given a *decomposition* by an expert, the inference *subproblems* are solved independently but communicate to each other by *passing messages*, that is, passing solutions. To implement this idea we use an approach which has a similar intuition as block coordinate decent (BCD) [32] methods, also referred to as alternating optimization (AO) [3]. In these methods, given a general objective function H of multivariate y, to find

the MAP of H we can divide the variables into a number of blocks assuming that each block has a local maximizer. Our suggested communicative inference is presented in algorithm 3 for a decomposition $\mathcal{S} = \{y_1, y_2\}$ containing two blocks of variables y_1 and y_2 for the objective H. At the starting point a random initializer assigns random values to the output variables in the decomposition set. Then at each step we optimize over one block of the target variables using off-the-shelf solvers while the other block is set with the last partial MAP assignment from the previous step. In contrast to the standard setting of AO, in our setting the variables are discrete and each subproblem is solved approximately by an LP-relaxation technique, in which the condition of having an integer solution is relaxed, and by using the relevant subset of constraints activated.

At prediction time, we establish communicative inference between two arbitrary models, each of which can be trained jointly or independently. At training time, it allows a joint update over all blocks in the weight vector. Hence it can provide more globally violating examples. The main advantage of this approach is that the models can follow their own Maximum a Posteriori estimation (MAP) methodology, based on any approximate or exact inference technique.

7. Model Specification

In this section, we formulate the problem of mapping natural language to spatial ontologies. We represent the supervised structured learning model designed for solving this problem using the Link-And-Label model described in Section 5 and specify: a) The input *components* and *types*; b) The output *single labels*, *linked labels* and *global constraints* over the output structure; c) The *joint feature templates*, *candidate generation* for the templates and the main *objective function*.

7.1. Input Space

The input part of each example, x, is originally a natural language sentence such as

"There is a white large statue with spread arms on a hill.",

and each sentence has a number of single components that are its contained *word*s. The single components of x in the above example are identified as $x = \{x_1, \ldots, x_{14}\}$, where x_i is the identifier of the ith word in the sentence. Each word in the sentence is described by a vector of the local features denoted by $\phi_{word}(x_i)$, e.g. `(There,EX,SBJ,. . .)` describing the word form, the part of speech, etc. There are also components composed of *pair*s and *triplet*s of words and their descriptive vectors are referred to as $\phi_{pair}(x_i, x_j)$ and $\phi_{triplet}(x_i, x_j, x_k)$. We define a number of relational features describing the relationships between words (e.g. distance). A feature vector of a composed component such as a pair, $\phi_{pair}(x_1, x_2)$ is described by the local features of x_1, x_2 and the relational features between them, `(before, 1)`. The details of the linguistic meaning and values of the applied features is explained in Section 3.2 and we refer back to it later in this section. A dummy word (x_{14} here) is added to the components of each sentence to be used for *undefined* roles.

7.2. Output Space

In the output space, an ontology $\mathcal{H}$ with Γ number of nodes is given. The nodes in the ontology are actually the target labels we tend to predict and denoted as $l_{target} = \{l_i | l_i \in \mathcal{H}, i = 0 \ldots \Gamma\}$. The ontology is defined as a set of labels where $(\mathcal{H}, \prec)$ is a partial order. The symbol $\prec$ represents the super-label relationship (see Section 5) thus

$$\forall l, l' \in \mathcal{H} : l \prec l' \text{ if and only if } l \text{ is a super-label of } l'.$$

The actual labels in the learning model are defined based on the nodes in the ontology (see Section 3.2 figure 1). We define one single label sp which is an indicator function that receives a word x_i and indicates whether it is a spatial indicator, denoted as $sp(x_i)$ or briefly as sp_i. The roles of trajector and landmark are defined as linked labels. We present them with label strings $sp.tr$ and $sp.lm$ and their indicator functions act on a pair of words. The link label $sp.tr(x_i, x_j)$ ($sp.lm(x_i, x_j)$) receives two ordered words and indicates whether the first word is a spatial indicator and the second is a trajector (landmark) with respect to the first; this also is denoted briefly as $sp_i.tr_j$ ($sp_i.lm_j$). We use two additional labels: nsp that indicates whether a word is not a spatial indicator, and $nrol$ that indicates whether a pair is neither $sp.tr$ nor $sp.lm$. These two labels are helpful to collect features of negative classes for distinguishing the spatial relations. The above mentioned labels are related to the SpRL semantic layer. The fine-grained semantics of spatial relations are indicated in lower nodes in the ontology related to the SpQL layer. All SpQL related nodes

are linked labels related to relations and the indicator function of each one identifies whether a spatial relation is of a certain spatial type such as *region*, *direction*, *EC*, and so on. We denote these linked labels by r_γ and the first one r_0 is the spatial relation label. These labels actually are linked labels that can be represented by label strings. For example, *sp.tr.lm.region* shows the single labels on which the *region* node in the ontology directly depends.

Output structure. The structural properties of the output are described in Section 3.1. Here, given the introduced parameters for the representation of the input and output in our LAL model, those constraints are formalized in Equations 7-15 (for description of each constraint with examples see the reference to the relevant section). The labels are represented as indicator functions for the candidate inputs. For example, $sp(x_i)$ is equal to one if the candidate word x_i is a spatial indicator. The first two additional constraints 7-8 associate the labels in the learning model to the nodes in the ontology. Constraints 9-12 are the *horizontal* constraints including *multilabel*, *spatial reasoning* and *composed-of* constraints. The last three formulas show the *vertical* constraints including the *is-a* constraint in Equation 13, the *null-assignment* constraint in Equation 14 and the *mutual exclusivity* constraint in Equation 15.

$$sp_i + nsp_i = 1 \tag{7}$$

$$sp_i tr_j + sp_i lm_j + sp_i nrol_j = 1 \tag{8}$$

$$sp_i tr_j - sp_i \leq 0, \quad sp_i lm_j - sp_i \leq 0 \qquad 3.1.(a) \tag{9}$$

$$sp_i - \sum_j (sp_i tr_j) \leq 0, \quad sp_i - \sum_j (sp_i lm_j) \leq 0 \qquad 3.1.(a) \tag{10}$$

$$sp_i tr_j + sp_i lm_j \leq 1 \qquad 3.1.(b) \tag{11}$$

$$\sum_i (sp_i tr_j) \leq 1, \quad \sum_i (sp_i lm_j) \leq 1 \qquad 3.1.(c) \tag{12}$$

$$sp_i tr_j lm_k r_{\gamma'} - sp_i tr_j lm_k r_\gamma \leq 0, \quad \forall \gamma \prec \gamma' \quad \gamma, \gamma' \in \mathcal{H} \qquad 3.1.(i) \tag{13}$$

$$\sum_{\gamma \in \mathcal{H}_{leafs}} r_\gamma(x_i, x_j, x_k) \geq r_0(x_i, x_j, x_k) \qquad 3.1.(ii) \tag{14}$$

$$\sum_{\gamma \in QSR_h} sp_i tr_j lm_k r_\gamma \leq 1, \quad \forall h, \quad \forall QSR_h \subset \mathcal{H}_{leafs} \qquad 3.1.(iii) \tag{15}$$

To clarify the notation in the last two constraints, as described in Section 2.2.2, in the lightweight ontology $\mathcal{H}$, we have three general types of spatial calculi models of, *region*, *direction* and *distance*. The leaf nodes in the ontology are constructed based on multiple spatial calculi. Here the set of leaf nodes is defined as $\mathcal{H}_{leafs} = QSR_{region} \cup QSR_{direction} \cup QSR_{distance}$. The *null-assignment* constraint 14 imposes at least one fine-grained semantic assignment in a leaf node when a spatial relation is predicted. In constraint 15, to express the mutual exclusivity we denote each group of leaf nodes that belong to a qualitative spatial representation model as QSR_h.

Since we use the pairwise links in the first layer, the full spatial relations are built using the following formula,

$$r_0(x_i, x_j, x_k) \leftarrow sp(x_i) \wedge sp.tr(x_i, x_j) \wedge sp.lm(x_i, x_k) \tag{16}$$

Output representation. The model should predict the labels of all input components. Hence, the output is the spatial ontology that is populated by the input components (i.e. segments of the input sentence). The populated ontology can be represented as a set of the sets of components associated with each label in the ontology. To illustrate, for the above mentioned example in Section 7.1, we represent the output example with the indicators with value one:

```
{ {sp(on)},{sp.tr(on,statue)},{sp.lm(on,hill)},{r0(on,statue,hill)},
  {region(on,statue,hill)},{direction(on,statue,hill)},
  {EC(on,statue,hill)},{above(on,statue,hill)} }
```

7.3. Joint Feature Mapping and the Main Objective Function

Templates. To describe the structure of the joint feature functions, we define the templates (see Section 5.3) in our model. We use four main types of templates: *Role-template*s, *Composed-of-template*s, *Is-a-template*s and *Negation-template*s.

- A *Role-template* connects an input component to a single label indicating the role of that component. We use a *Role-template*, for instance, for spatial indicators denoted as *word.sp*. The input type of this template is a *single word.*

- A *Composed-of-template* connects a composed input component to a linked label. A linked label in this type of template contains sub-labels where linking them constructs new complex parts of the output. We mainly use two main *Composed-of-templates* to connect trajectors/landmarks and spatial indicators. These templates indicate whether a pair of words have the *trajector-of* or *landmark-of* relationship and compose a part of a *spatial relation*. We denote them as *pair.tr*, *pair.lm*. We define an additional *Composed-of-template* which is more complex and connects the three labels trajector, landmark and spatial indicator. This template indicates whether three words compose a spatial relation and it is denoted as $triplet.r_0$.

- An *Is-a-template* connects a single or a composed component to a linked label. The linked label contains sub-labels that have is-a relationships to each other.

  For all semantic types of spatial relations we use such a template that connects spatial relations to their spatial relationship semantics, such as regional, directional, etc. We show them as $triplet.r_\gamma$ indicating the type of input that is *triplet*, and the semantic label r_γ linked label. It connects the *spatial relation* type to the more fine-grained spatial semantics.

- A *Negation-template* indicates when a single or linked label (referring to a single or composed concept in the output) is not assigned as one. We use two *Negation* templates. The first template regards a single label for non-spatial indicators denoted as *word.nsp*. The second contains a negative linked label that indicates when a word is not a trajector nor a landmark with respect to a spatial indicator candidate. This *Negation-template* is denoted as *pair.nrol*.

Now we specify the candidate generators and features of these templates.
Candidate generators. The spatial indicators are prepositions in our model which are mostly tagged as IN and TO by parsers. Hence, we prune their candidates based on the POS-tags. Since prepositions belong to a closed lexical category, we collect a lexicon for prepositions according to our corpus. For the roles of trajector and landmark also a subset of words is selected. We define three basic sets of useful words in our problem as,

$$\begin{aligned} C_1 &= \{x_i | Pos(x_i) \in \{IN, TO\} \vee x_i \in PrepositionLexicon\}. \\ C_2 &= \{x_i | Pos(x_i) \in \{NN, NNS\} \vee Dprl(x_i) = SBJ \vee x_i = undefined\}, \\ C_3 &= \{x_i | Pos(x_i) \in \{NN, NNS, PRN\} \vee x_i = undefined\}, \end{aligned} \tag{17}$$

and choose the candidates for the labels based on these as follows:

$$C_{nsp} = C_{sp} = C_1, \;\; C_{sp.tr} = C_{sp} \otimes C_2, \;\; C_{sp.lm} = C_{sp} \otimes C_3, \; C_{nrol} = C_2 \cup C_3, \;\; C_{r_\gamma} = C_{sp} \otimes C_2 \otimes C_3,$$

where each C_{label} denotes the set of candidates for a *label*, $Pos(x_i)$ is a function that returns the POS-tag of a word x_i and $Dprl(x_i)$ returns the label assigned by the dependency parser to a word x_i. *PrepositionLexicon* is the collected list of possible prepositions according to the available corpus. For the trajectors, the roles are assigned to singular (NN) or plural nouns (NNS) or the words that are labeled as subject (SBJ) in the dependency tree. For the landmarks the roles are assigned to singular, plural or proper nouns (PRN). When the dataset contains dynamic spatial relations, that is, relations that indicate motions, trajector candidates can also have the POS label PRP or WRB. Moreover, *undefined* can be a candidate for trajector and landmark roles. The linguistic features are extracted by the syntactic and the dependency parser.
Input feature functions. The input part of each template is characterized by a binary input feature vector, which is produced based on the local and relational features of the input components. We denote this feature vector using the symbol ϕ indexed by the relevant label,

$\phi_{sp}(x_i) \triangleq \phi_{nsp}(x_i) \triangleq$ *local features of the word* x_i.

$\phi_{sp.tr}(x_i, x_j) \triangleq \phi_{sp.lm}(x_i, x_j) \triangleq \phi_{sp.nrol}(x_i, x_j) \triangleq$ *local features of the word* x_j, *relational features of the pair* x_i *and* x_j.

$\phi_{triplet_{r_\gamma}}(x_i, x_j, x_k) \triangleq$ *local features of* x_i, x_j, x_k, *relational features of the pair* x_i, x_j *and, the pair* $x_i, x_k \;\; \forall r_\gamma \in \mathcal{H}$.

The local and relational features are described in Section 3.2.

Link-And-Label objective. Each instantiation of a template represents a joint feature sub-mapping. It is calculated by the product of a vector of the input features and an output label which is a single valued binary variable. For example, $\phi_{sp_i}.sp_i$ refers to the input features of the ith spatial indicator candidate multiplied by the value of sp_i, and $\phi_{sp_itr_j}.sp_itr_j$ refers to the features of the ith spatial indicator candidate with respect to the jth trajector candidate multiplied by the label $sp_i.tr_j$. We capsulate these two parts in a joint feature function f_p, associated with each template C_p with a label and its relevant input. We represent these, for example, as $f_{sp}(sp_i)$ and $f_{sptr}(sp_itr_j)$. In this joint feature function, the label name and the index make the necessary connection to the input candidate. There is no need to show the x component explicitly. This function will be a zero vector if the label of the candidate is zero and will be equal to the input features of the candidate if the label is one. We have a block of weights for each template in the target model as,

$$W = [W_{sp}, W_{nsp}, W_{sptr}, W_{splm}, W_{spnrol}, W_{r_0}, \ldots, W_{r_\Gamma}].$$

To construct the objective function $g = \langle W, f(x, y)\rangle$, each candidate for each label that is generated according to a template specification, is mapped to a joint feature vector (referred to as local joint feature). The local joint feature function is associated with a block W_p of weights for that template. In fact, the parameters of the variables related to one template are tied. The objective function is a linear function of feature values implying that we should sum over all the produced feature vectors multiplied by their weights,

$$\begin{aligned}
&\langle W_{sp}, f_{sp}(sp_1)\rangle + \cdots + \langle W_{sp}\, f_{sp}(sp_{SP})\rangle + \langle W_{nsp}, f_{nsp}(nsp_1)\rangle + \cdots + \langle W_{nsp} f_{nsp}(nsp_{SP})\rangle + \\
&\langle W_{sptr}, f_{sptr}(sp_1tr_1)\rangle + \cdots + \langle W_{sptr}, f_{sptr}(sp_1tr_{TR})\rangle + \\
&\qquad\qquad \ddots \\
&\langle W_{sptr}, f_{sptr}(sp_{SP}tr_1)\rangle + \cdots + \langle W_{sptr}, f_{sptr}(sp_{SP}tr_{TR})\rangle + \langle W_{splm}, f_{splm}(sp_1lm_1)\rangle + \cdots + \langle W_{splm}, f_{splm}(sp_1lm_{LM})\rangle + \\
&\qquad\qquad \ddots + \langle W_{r_\Gamma}, f_{sptrlmr_\Gamma}(sp_{SP}tr_{TR}lm_{LM}r_\Gamma)\rangle.
\end{aligned} \tag{18}$$

During prediction when finding the best assignments, we rewrite the f_p local joint feature functions as the product of input feature functions ϕ and the unknown output labels. We obtain a function g in terms of the labels. We can rewrite and represent the instances of the same template, which are associated with the same block of the weight vector, compactly as,

$$\begin{aligned}
&\sum_{i\in C_{sp}} \langle W_{sp}, \phi_{sp_i}\rangle \cdot sp_i + \sum_{i\in C_{sp}} \langle W_{nsp}, \phi_{nsp_i}\rangle \cdot nsp_i + \sum_{i\in C_{tr}} \sum_{j\in C_{sp}} \langle W_{sptr}, \phi_{sp_itr_j}\rangle \cdot sp_itr_j + \sum_{i\in C_{lm}} \sum_{j\in C_{sp}} \langle W_{splm}, \phi_{sp_ilm_j}\rangle \cdot sp_ilm_j + \\
&\sum_{i\in C_{nrol}} \sum_{j\in C_{sp}} \langle W_{nrol}, \phi_{sp_inrol_j}\rangle \cdot sp_inrol_j + \sum_{\gamma=1}^{\Gamma} \sum_{i\in C_{sp}} \sum_{i\in C_{tr}} \sum_{i\in C_{lm}} \langle W_\gamma, \phi_{sp_itr_jlm_kr_\gamma}\rangle \cdot sp_itr_jlm_kr_\gamma.
\end{aligned} \tag{19}$$

This is the objective function that we need to maximize in the prediction time in order to find the best label assignments considering the global constraints mentioned in Section 7.2. We refer to the four terms in Equation 19 as the F_{SpRL} and the last term as the F_{SpQL}. The constraints can be between the variables related to one template, for example, the counting constraints over spatial indicators; or can be formulated more globally across templates, for instance, by the *spatial reasoning* constraints.

7.4. Component Based Loss

As described in Section 4, in structured training models we need to find the most violated output in Equation 2 for each training example per training iteration. Moreover, as mentioned in Section 5, to have a loss which factorizes similar to the feature function, we define a component-based loss for each label l according to Equation 6. We write a loss by considering all the instantiated labels according to the templates as follows,

$$\Delta_{sp}(\Lambda_{sp}, \Lambda'_{sp}) = \sum_{i \in C_{sp}} (1 - 2sp'_i) sp_i + \sum_{i \in C_{sp}} sp'_i \tag{20}$$

$$\Delta_{nsp}(\Lambda_{nsp}, \Lambda'_{nsp}) = \sum_{i \in C_{sp}} (1 - 2nsp'_i) nsp_i + \sum_{i \in C_{sp}} nsp'_i$$

$$\Delta_{sptr}(\Lambda_{sptr}, \Lambda'_{sptr}) = \sum_{i \in C_{sp}} \sum_{j \in C_{tr}} (1 - 2sp'_i tr'_j) sp_i tr_j + \sum_{i \in C_{sp}} \sum_{j \in C_{tr}} sp'_i tr'_j$$

$$\Delta_{splm}(\Lambda_{splm}, \Lambda'_{splm}) = \sum_{i \in C_{sp}} \sum_{j \in C_{lm}} (1 - 2sp'_i lm'_j) sp_i lm_j + \sum_{i \in C_{sp}} \sum_{j \in C_{lm}} sp'_i lm'_j$$

$$\Delta_{spnrol}(\Lambda_{spnrol}, \Lambda'_{spnrol}) = \sum_{i \in C_{sp}} \sum_{j \in C_{nrol}} (1 - 2sp'_i nrol'_j) sp_i nrol_j + \sum_{i \in C_{sp}} \sum_{j \in C_{nrol}} sp'_i nrol'_j$$

$$\Delta_{r_\gamma}(\Lambda_{r_\gamma}, \Lambda'_{r_\gamma})) = \sum_{\gamma=1}^{\Gamma} \sum_{i \in C_{sp}} \sum_{i \in C_{tr}} \sum_{i \in C_{lm}} \omega_\gamma (1 - 2r'_\gamma sp'_i tr'_j lm'_k) sp_i tr_j lm_k r_\gamma + \sum_{\gamma=1}^{\Gamma} \sum_{i \in C_{sp}} \sum_{i \in C_{tr}} \sum_{i \in C_{lm}} \omega_\gamma sp'_i tr'_j lm'_k r'_\gamma.$$

In fact, we count all the wrong label assignments which have been made for all the template instantiations (see Hamming loss Equation 5). The first five lines in Equation 20 are related to the SpRL labels which are averaged and aggregated with the loss of spatial semantic labels related to the SpQL layer. In the SpQL part, the *preferences* ω_γ of the labels r_γ inversely depend upon the distance of that label node from the r_0 node in the spatial ontology. The nodes closer to the leaves are assigned a lower value. This implies that we give a higher priority to the classification of more general semantics than the fine-grained spatial semantics in the leaf nodes of the tree. This is very straightforward and corresponds to the semantics of our problem. More specifically, we set the preferences such that firstly the siblings with a common parent have a similar preference, secondly the preference of the parent is two times larger than the preference of its children, and thirdly the sum of all the preferences in the ontology is equal to one, to obtain a loss value between 0 and 1. Given these conditions, there is a unique way to assign preferences to labels in the ontology.

8. Local-Global Training and Prediction Models

In this section we collect the required pieces from the last sections and discuss the model variations belonging to the spectrum of local and global training and prediction models that we design.

The global loss augmented objective function of our problem is built by adding the components of Equations 19 and 20. We train the parameters W of the function g in the framework of discriminative inference-based structured prediction models such as structured SVM, structured perceptron and average perceptron algorithms. To train the parameters W including all W_p blocks of weights jointly, we need to maximize the constructed objective function based on the defined templates globally. This yields the most violated outputs for each training example per training iteration. This MAP problem over loss-augmented g containing both semantic layers of SpRL and SpQL is computationally highly complex. Moreover, the solution to this inference should fulfill the structural constraints discussed in Section 7.2, which makes it even harder. The initial feasible output space contains the space of all possible spatial relations multiplied by all possible ontological semantic assignments for each spatial relation, that is $O(n^3 \times 2^\Gamma)$ where n is the number of candidate words per sentence, which is assumed to be in the order of the length of the sentence n and Γ is the number of nodes in the ontology. Optimizing our proposed formulation for the objective function encompassing the variables in the two layers is also computationally complex. The complexity is due to the large number of $sp_i tr_j lm_k r_\gamma$ variables, which is $O(n^3 * \Gamma)$ in this formulation, given that for each i, j, k all the $r_1..r_\gamma$ variables are also correlated and should fulfill the ontological constraints. To solve this as a combinatorial optimization problem using off-the-shelf solvers is still challenging. We consider this objective as a linear function and provide a linear formulation of the constraints 7-15 and solve it with LP-relaxation techniques. The global constraints 7-12 involve the variables of the SpRL layer and help to exploit the internal structure of the relations and their global correlations in the sentence. The constraints are described in section 7.2. The ontological constraints over the semantic assignments represented in the constraints 13-15 involve the variables of the SpQL layer.

For the experiments, first we design various models per semantic layer. Afterwards, we build a global model for both layers which necessitates going beyond applying off-the-shelf solvers. We use our proposed communicative inference approach for the global optimization over both layers. In the models listed below we increase the *globality* in the training and prediction gradually. By globality we literally mean the number of output variables that are considered collectively/jointly during training and prediction [33].

- **LO setting per layer.** A basic model, which can still be practical depending on the application, is the learning only (LO) model. In the LO setting, independent binary classifiers are built for each single/linked label in the SpRL and SpQL layers referred to as **LOSpRL** and **LOSpQL** respectively. These local models make independent binary predictions per label.

- **(L+I) setting per layer.** In the learning plus inference (L+I) setting, locally trained models are used, but the joint prediction is performed by constrained optimization of the objective function shown in Equation 19 subject to the constraints 7-15. The main prediction time objective function is split into two parts, each of which relates to one layer with its own independent constraints. We refer to these models as **LISpRL** and **LISpQL** for the SpRL and SpQL layers.

- **IBT setting per layer.** In the inference-based-training setting, the objective of the loss-augmented inference is solved during training per layer. In other words, it is split into two parts, each part containing its own loss function and considering its own independent constraints. These models are referred to as **IBTSpRL** and **IBTSpQL**, for the SpLR and SpQL respectively.

In the above mentioned models for learning and prediction in the second layer we assume the ground truth of the first layer of spatial relations are available and the focus of the training and prediction is on the fine-grained semantics of the relations. This assumption is useful for analyzing the difficulties of the two layers independently in the experimental section. However, we need to connect the two layers and make global training and prediction models encompassing both layers. Therefore the following models are designed:

- **L+I setting joining two layers.** In this model, we use the above mentioned IBT models per layer but make a global optimization jointly for both layers during prediction time. We use *communicative inference* (Algorithm 3) during prediction while solving each sub-problem using constraint optimization and LP-relaxation. We call this model **EtoE-IBTCP**.

- **IBT setting joining two layers.** To train a model jointly for both layers, we use *communicative inference* (Algorithm 3), which connects the two combinatorial subproblems in the training time for solving the global loss augmented inference. This globally trained model makes a joint prediction for the two layers too. We call this model **EtoE-IBTCTCP**.

We compare the global IBT model covering both layers to the pipeline of the IBT models of each layer in the **EtoE-pipe** model.

9. Experiments

For the extraction of the linguistic features we use the LTH[2] tool that produces features in the CoNLL-08 format[3]. The applied machine learning techniques are the structured SVM using the svm-struct Matlab wrapper [34] (coded as **SSVM**) and our implementation of the structured perceptron (coded as **SPerc**) and the averaged structured perceptron (coded as **AvGSPerc**). For local learning settings a binary SVM (coded as **BSVM**) is used. For the LP-solver, we used the Matlab optimization tool (specifically, the *bintprog* program).
Dataset. We use the SemEval-2012 and SemEval-2013 shared tasks data referred to as CLEF and Confluence respectively.

[2] http://barbar.cs.lth.se:8081/
[3] http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:format

Lables	Training-Evaluation
sp	1466
tr	1588
lm	1184
*r*0	1706

Table 1. Spatial role statistics of the SemEval-2012 dataset, which contains 1213 sentences and 20,095 words.

Spatial relations	1706				
Topological	EQ	DC	EC	PO	PP
1040	6	142	462	15	417

Directional	BELOW	LEFT	RIGHT	BEHIND	FRONT	ABOVE
639	18	159	103	101	185	71

Distance
82

Table 2. The number of occurrences of the added semantic labels in the SemEval-2012 dataset.

	tr	*lm*	*sp*	*mi*	*path*	*di*r	*dis*	*r*0
Training	1701	1037	879	1039	945	223	307	2105
Evaluation	497	316	247	305	240	37	87	598

Table 3. Corpus statistics for SpRL-2013 with respect to annotated spatial roles (trajectors (tr), landmarks (lm), spatial indicators (sp), motion indicators (mi), paths (path), directions (dir) and distances (dis)) and spatial relations(r0), *mi* indicates the number of dynamic relations.

CLEF. The SemEval-2012 corpus[4] [35] consists of textual descriptions of 613 images originally selected from the IAPR TC-12 dataset [36], provided by the *CLEF* organization. Table 1 and Table 2 respectively show the statistics of the SpRL layer and of the SpQL layer annotations. The primary focus of the experiments is on this CLEF dataset which has been augmented with the annotations of the SpQL layer as described in Section 2.

Confluence. The SemEval-2013 corpus[5] consists of 1789 sentences that describe world locations. Table 3 shows the statistics of the SpRL annotations of this corpus. In contrast to the CLEF texts that describe only static spatial relations, this corpus also contains narrative and dynamic spatial information. The statistics of this dataset show that each motion indicator participates in forming one spatial relation at maximum. If we assume that the relations which include a motion indicator, are dynamic relations, then about 50% of the relations in Confluence are dynamic. Unfortunately this data is not yet annotated with the SpQL information. In our experiments it is used as an additional corpus for evaluating the recognition of the SpRL layer.

Evaluation. We use classic machine learning evaluation metrics. The evaluations on CLEF are based on 10-fold cross validation. For Confluence we used the proposed train/evaluation setting of the shared task. In all the designed learning models the evaluations are provided per node in the ontology if relevant and predicted for that specific model. More precisely, we evaluate the predictions for the single labels including the trajectors, landmarks and spatial-indicators and for the pairwise linked labels including spatial indicator-trajector and spatial indicator-landmark. Moreover, the evaluation is performed for the triplets of spatial relation and their semantic type in the ontology, such as region, etc. The evaluation metrics of precision, recall and F1 measure are used, and are defined as:

$$recall = \frac{TP}{TP + FN}, \quad precision = \frac{TP}{TP + FP}, \quad F1 = \frac{2 * recall * precision}{(recall + precision)}$$

where TP is the number of predicted components that exactly match the ground truth, FP is the number of predicted components that do not match the ground truth and FN is the number of ground truth components that do not match the predicted components.

In the evaluation of linked labels, a prediction is true when all the composing single labels accord with the ground truth. These values are counted per test sentence and are summed up over all the sentences in the test set for each fold. The precision, recall and F1 are calculated for each fold separately and afterwards averaged over the 10-folds (i.e. macro-averaging) per label. The evaluations will be reported based on the performance of models over the two layers. For the SpRL layer each label and linked label are reported separately. For the SpQL layer each linked label is evaluated separately and then the weighted averages over all SpQL linked labels in the ontology are reported. Because the number of examples is highly variable among SpQL linked labels, each metric value for a SpQL linked label is weighted with the proportion of its examples when calculating the final value of each evaluation metric (micro-averaging). In the CLEF dataset all words are labeled with their respective semantics, in the Confluence dataset

[4] http://www.cs.york.ac.uk/semeval-2012/task3/

[5] http://www.cs.york.ac.uk/semeval-2013/task3/

only headwords of sentence constituents obtained with a dependency parse are labeled. In the Confluence dataset we conflate the labels of motion indicators, direction and distance indicators to the class of spatial indicators, and assume the path labels as landmarks. In this way the obtained label structure fits the SpRL layer of our spatial relation recognizer.

9.1. Experimental Research Questions

We provide an empirical investigation of the efficiency and the performance of the designed structured learning models for mapping natural language to spatial ontologies. The experiments are organized based on the evaluation and comparison of LO, L+I and IBT learning schemes, and after all our global model based on communicative inference. The main research question is:

Q. *What is the influence of increasing the globality of training and prediction for the spatial ontology population?*

We investigate this question by examining the results per layer and the results for the whole ontology, while answering the following questions.

Q1. What is the performance of the local models which make local predictions in the SpRL layer *(i.e. LOSpRL evaluation)*?

Q2. What is the performance of an LO model for the SpQL layer given the spatial relations (*i.e. LOSpQL evaluation*)?

Q3. Does global prediction in the L+I setting improve the results of SpRL *(i.e. LISpRL vs. LOSpRL)*?

Q4. Does the global prediction in the L+I setting improve the results of SpQL *(i.e. LISpQL vs. LOSpQL)*?

Q5. Does considering correlations among labels in the IBT model improve the results of SpRL *(i.e. IBTSpRL vs. LISpRL)*?

Q6. Does considering correlations among labels in the IBT model improve the results of SpQL *(i.e. IBTSpQL vs. LISpQL)*?

Q7. What is the performance of connecting the two IBT models trained independently for the two layers and making the prediction in a pipeline *(i.e. EtoE-pipe evaluation)*?

Q8. Can we use the above mentioned model, but practically make a global prediction over both layers? Can the communicative inference algorithm help to make this global prediction *(i.e. EtoE-pipe vs. EtoE-IBTCP)*?

Q9. Can we practically train a global model having a global loss-augmented inference and jointly train for the two layers? Can the communicative loss-augmented inference help to achieve such a global model over the two layers *(i.e. EtoE-pipe vs. EtoE-IBTCTCP)*?

Q10. Can we apply the proposed models on text containing dynamic spatial information? Does the answer to the above relevant questions still hold when applying them?

9.2. Local Learning Local Prediction

In the local models LOSpRL and LOSpQL we train for the templates (see Section 7.3) of the target model independently. The target output is built based on local binary predictions. The number of examples for each local model is equal to the number of candidates generated by the candidate generator of its template. It implies the candidates are assumed as i.i.d examples although the ones extracted from one sentence are certainly not.
The results are reported in table 4. The candidate selection (see Section 7.3) causes between 2%-3% missed positives leading to a corresponding drop in recall. Using relational features of pairs in this experiment helps to learn meaningful local models to predict the link between *sp* with *tr* and *sp* with *lm*. We produce the instances of the spatial relation node (r_0) using rule 16, yielding F1=0.503 for this linked label.

Due to the large number of negative (not spatial) relations in the training sentences (193,890), the local learning for the *sp.tr.lm* linked labels which is actually the binary classification of all possible relations does not provide

Target	Precision	Recall	F1	Annotated	Positive candidates[a]	Negative candidates
sp	0.875	0,944	0,907	1466	1437	1992
sp.tr	0.776	0.590	0.668	1693	1640	20133
sp.lm	0.868	0.793	0.827	1196	1161	24123
r_0	0.498	0.510	0.503	1703	1619	–

Table 4. LO (LOSpRL): Local training, local prediction for single label *sp*, linked labels *sp.tr*, *sp.lm* and producing r_0 using rule 16, BSVM.

[a]The number of positive candidates is less than the actual annotations due to the candidate selection discussed in section 7.2.

Target	Precision	Recall	F1	Annotated	Positive candidates	Negative candidates
sp	0.881	0,942	0,909	1466	1437	1992
sp.tr	0.752	0.622	0.678	1693	1640	20133
sp.lm	0.853	0.815	0.832	1196	1161	24123
r_0	0.526	0.533	0.529	1703	1619	–

Table 5. L+I (LISpRL): Local training, global prediction for single label *sp*, linked labels *sp.tr*, *sp.lm* and producing r_0 using rule 16, BSVM.

meaningful results using the basic BSVM. Hence, to be able to learn the linked labels in the SpQL layer in a local learning setting, we did experiments by assuming that the ground truth relations are given. Table 6 shows these results using BSVM.

Discussion: Local training in both layers would mean training basic relation classifiers to classify the spatial relations to indicate whether they carry specific spatial semantic information or not. This setting can be useful if more combinatory contextual features based on the training data are used and some stronger heuristics for candidate pruning are employed. This indicates the necessity of designing models that are able to exploit more abstract structural features. These experiments clarify the answer to the research questions **Q1** and **Q2**. For the SpQL semantic labels, if the ground truth relations are given, then local learning can yield reasonable results particularly for the labels with a larger number of positive examples, see table 6.

9.3. Local Learning Plus Inference

In the second experimental setting we use the same locally trained models using BSVM but build the global prediction for the two layers. The LISpRL model in table 5 is locally trained for the SpRL layer and does global prediction over *sp*, *sp.tr*, and *sp.lm* using the constraints 7-12 and the related part of the objective function. The LISpQL model populates the nodes of SpQL layer. This model receives the ground-truth relations and uses locally trained binary classifiers for each semantic type. It performs global prediction and imposes the ontological constraints 13-15 for the SpQL layer. Table 7 shows the results of this experiment. The extensive analysis of the effect of the individual constraints for the two layers is given in [18].

Discussion: Table 5 shows that exploiting the structure of the output via a global constraint optimization during prediction increases the performance of labeling spatial relations in the SpRL layer and also the qualitative types in the SpQL layer compared to the local predictions in table 4. For the SpQL labels (when using ground truth relations), as expected, imposing the constraints decreases the number of false positives, leading to an increase in the overall weighted average of precision and a drop in recall. However the overall F1 measure increases by 0.007 when compared to the local predictions in table 6 (significant for p=0.1)[6]. In the nodes EC, PP, PO, BELOW, RIGHT, BEHIND, FRONT and ABOVE a dramatic increase is visible. Only in the two nodes of LEFT and DC, there is a drop in F1. In these two nodes though the precision increases, the decrease in recall is comparatively larger. Distinguishing between LEFT and RIGHT is difficult for our model since the features are often similar except for the lexical form.

The results indicate that imposing the constraints sharply increases the performance of the lower level nodes in the ontology compensating for the lack of examples for those nodes. These results clarify the answers to the questions **Q3** and **Q4**.

9.4. Inference Based Training

[6]We used the t-test for all reported significance tests.

Class	Precision	Recall	F
Region	0.943	0.893	0.916
Direction	0.842	0.919	0.876
Distance	0.114	0.835	0.198
EQ	0.3	0.7	0.1
DC	0.407	0.625	0.45
EC	0.533	0.842	0.631
PO	0.014	0.607	0.0255
PP	0.568	0.841	0.671
BELOW	0.68	0.76	0.597
LEFT	0.370	0.961	0.504
RIGHT	0.108	0.992	0.187
BEHIND	0.492	0.976	0.636
FRONT	0.234	0.964	0.363
ABOVE	0.680	0.771	0.696
W.Avg.	0.6918	0.877	0.735

Table 6. LO (LOSpQL): Given G-truth relation, local training, local prediction for labels $r_{1..O}$, BSVM.

Class	Precision	Recall	F
Region	0.939	0.928	0.933
Direction	0.693	0.939	0.785
Distance	0.114	0.835	0.198
EQ	0.3	0.7	0.1
DC	0.608	0.356	0.386
EC	0.656	0.7515	0.692
PO	0.7	0.479	0.370
PP	0.771	0.747	0.754
BELOW	0.867	0.81	0.722
LEFT	0.561	0.297	0.330
RIGHT	0.204	0.831	0.310
BEHIND	0.748	0.918	0.811
FRONT	0.789	0.920	0.833
ABOVE	0.725	0.771	0.726
W.Avg.	0.739	0.815	0.742

Table 7. L+I (LISpQL): Given G-truth relation, local training, global prediction, constraints 13- 14 over labels $r_{1..O}$, BSVM.

Class	Precision	Recall	F
Region	0.936	0.951	0.943
Direction	0.891	0.918	0.903
Distance	0.820	0.785	0.790
EQ	0.9	0.7	0.6
DC	0.596	0.603	0.582
EC	0.724	0.780	0.747
PO	1	0.529	0.544
PP	0.7807	0.793	0.783
BELOW	0.8167	0.76	0.672
LEFT	0.518	0.755	0.553
RIGHT	0.517	0.333	0.349
BEHIND	0.920	0.902	0.903
FRONT	0.838	0.897	0.859
ABOVE	0.846	0.821	0.813
W.Avg.	0.822	0.844	0.821

Table 8. IBT (IBTSpQL): G-truth relation, best Global learning-Global prediction for SpQL, SSVM.

Class	Precision	Recall	F
Region	0.667	0.541	0.594
Direction	0.602	0.544	0.57
Distance	0.633	0.409	0.477
EQ	0.9	0.7	0.6
DC	0.383	0.304	0.330
EC	0.571	0.442	0.486
PO	0.85	0.464	0.458
PP	0.577	0.48	0.521
BELOW	0.6	0.55	0.49
LEFT	0.449	0.292	0.331
RIGHT	0.372	0.537	0.359
BEHIND	0.602	0.563	0.573
FRONT	0.558	0.508	0.525
ABOVE	0.654	0.485	0.513
W.Avg	0.593	0.493	0.527

Table 9. Pipeline SpRL and SpQL (EtoE-pipe): AvGSPerc.

In this part of experiments we perform loss-augmented inference using LP-solvers to train global models in each layer separately. The experiments are performed using both SSVM and AvGSPerc for structured training. To summarize the results, we show only the best results of the two layers which is obtained by AvGSPerc for the first layer shown in Table 10 and by SSVM for the second layer shown in Table 8. AvGSPerc is sharply better than the basic structured perceptron model (F1 (SpRL)=0.574 and F1(SpQL)=0.757) and outperforms SSVM also for the SpRL layer (SSVM(SpRL), F1 ($r0$)=0.579). But when using the ground truth relations for the SpQL layer, the SSVM performs better (AvGSPerc(SpQL) F1 of weighted averaging = 0.791 compared to Table 8). These results provide the answers to the questions **Q5** and **Q6**.

Target	Precision	Recall	F1
sp	0.905	0.8416	0.871
sp.tr	0.728	0.610	0.662
sp.lm	0.828	0.766	0.794
r_0	0.663	0.554	0.602

Table 10. IBT (IBTSpRL): Best Global learning-Global prediction over *sp*, *sp.tr*, *sp.lm* (using nsp, sp-nrol) building r_0 using rule 16 AvGSPerc.

Discussion: IBTSpRL and IBTSpQL models for SpRL and SpQL make more realistic i.i.d. assumptions and consider the correlations between all relations in one sentence. By picking global violating examples for each layer, we can deal with the huge number of negative relations and exploit the structure of the output in training the model parameters.

The global training for each layer shows an increase in the performance compared to local training. In the IBTSpRL model, although the individual roles are recognized with a lower performance, the spatial relation formed by the roles is recognized more accurately. We observed that using the SpQL constraints 13-15 for finding the most violated *y* provides a small improvement compared to doing inference without the constraints, but the convergence of the training is faster in the former case. Our analysis shows that this is due to taking the more elegant wrong *y*s that respect the structure of the output. Imposing the constraints makes the LP-solver slower but the number of iterations in the quadratic optimization procedure for updating the weights by SVM-struct decreases and overall the training is 10 times faster when using constraints 13-15. Another finding about global learning models IBTSpRL and IBTSpQL is that when the constraints are applied during training, applying them during prediction is required to obtain an accurate prediction. This is expected because when using constraints during training, the trained model relies on the structure of the output in addition to the weight vectors to distinguish between labels. Consequently the absence of the constraints for prediction time leads to an inaccurate prediction (F1=0.264).

9.5. End-to-End Decomposition and Communication

Although there are efficient off-the-shelf solvers to be used for solving the required inference phases in our model, the target global SpRL-SpQL machine learning model does not scale up due to the large number of candidate relations and large number of constraints. Hence in the following sections we experiment with different solutions for building the end-to-end SpRL and SpQL, and particularly evaluate the proposed communicative inference approach when it is used during training and prediction.

Pipeline. A straightforward approach is to use the separately trained models of IBTSpRL and IBTSpQL in a pipeline for prediction. We refer to this model as EtoE-pipe. In EtoE-pipe the same IBTSpRL predicts the first layer and then the prediction is piped to IBTSpQL. The best results of this model are achieved based on AvGSPerc training and they are shown in tables 9 and 10 for the two layers. These results provide the answers to the questions **Q7**.

Communicative prediction. In this experiment we exploit the IBTSpRL and IBTSpQL models and apply the suggested algorithm of communicative inference (see Algorithm 3) during prediction. This model is referred as EtoE-IBT-CP.

We performed experiments with this model using both SSVM and AvGSPerc for training. Once more the overall results of AvGSPerc were 3% better than SSVM in F1 measure. The best results are presented in tables 11 and 13 for the two layers. The results on SSVM show about 0.001 improvement for SpQL and about 0.01 (significant for $p = 0.1$) improvement over the SpRL when it receives feedback from SpQL during prediction compared to the EtoE-pipe model. The results on AvGSPrec for the SpRL and SpQL are consistently outperforming (~ 0.003) compared to the above mentioned pipeline model; the improvement for SpRL is not strongly significant but for SpQL is significant for $p = 0.1$. These results provide the answers to the questions **Q8**.

Communicative training. The communicative training is experimented with the best-performing learning technique, which is AvGSPerc according to the above experiments. This model is referred as EtoE-IBT-CLCP and is trained over SpRL and SpQL layers jointly via communicative inference during training and makes global predictions based

Class	Precision	Recall	F1
Region	0.667	0.542	0.594
Direction	0.616	0.547	0.578
Distance	0.67	0.409	0.488
EQ	0.9	0.7	0.6
DC	0.378	0.304	0.329
EC	0.571	0.441	0.485
PO	0.85	0.464	0.458
PP	0.582	0.482	0.525
BELOW	0.6	0.55	0.49
LEFT	0.452	0.292	0.332
RIGHT	0.377	0.544	0.365
BEHIND	0.611	0.563	0.576
FRONT	0.565	0.508	0.531
ABOVE	0.654	0.485	0.513
W.Avg.	0.598	0.495	0.529

Table 11. SpQL by communicative prediction SpRL-SpQL (EtoE-IBT-CP): AvGSPerc.

Class	Precision	Recall	F1
Region	0.627	0.545	0.581
Direction	0.618	0.574	0.592
Distance	0.513	0.323	0.345
EQ	0.9	0.7	0.7
DC	0.238	0.238	0.218
EC	0.506	0.393	0.433
PO	0.85	0.414	0.392
PP	0.495	0.502	0.495
BELWO	0.558	0.575	0.38
LEFT	0.467	0.255	0.188
RIGHT	0.339	0.590	0.373
BEHIND	0.607	0.568	0.582
FRONT	0.562	0.533	0.538
ABOVE	0.544	0.503	0.518
W.Avg.	0.553	0.491	0.500

Table 12. SpQL by communicative training and prediction SpRL-SpQL (EtoE-IBT-CLCP), AvGSPerc.

Target	Precision	Recall	F1
sp	0.907	0.838	0.870
sp.tr	0.732	0.610	0.663
sp.lm	0.831	0.764	0.795
r_0	0.669	0.556	0.605

Table 13. SpRL by communicative prediction by SpRL-SpQL (EtoE-IBTCP): AvGSPer.

Target	Precision	Recall	F1
sp	0.905	0.84	0.869
sp.tr	0.733	0.625	0.673
sp.lm	0.831	0.769	0.797
r_0	0.673	0.573	0.617

Table 14. SpRL by communicative training and prediction by SpRL-SpQL(EtoE-IBT-CLCP): AvGSPerc.

on the same communicative approach. The performance of this model is presented in tables 12 and 14. These results show that the communication during training improves the performance of IBTSpRL in table 13 (significant for $p = 0.1$). This is due to the provided feedback about the type of the spatial relations to the first layer for recognizing the spatial roles. However the performance of the second layer dropped in this setting compared to EtoE-IBT-CP (see table 11) which does communication only during prediction. This behavior can be due to the high performance of the SpQL layer in general (see table 8) compared to the SpRL layer, therefore the feedback from the semantic types can promote the role labeling, while learning from the noisy role labels in the presence of a small dataset is more tricky for recognizing the spatial qualitative labels. These results provide the answers to question **Q9**.

Efficiency analysis. The AvGSPerc is highly efficient compared to SSVM. Though the cutting plane algorithm in SSVM reduces the duration of each training iteration, it requires many more iterations than the AvGSPerc according to our experiments. We achieve the best models in 10-20 iterations for AvGSPerc while for SSVM by setting the training error to less than 0.1 at least 80 iterations are needed to converge. In table 15 the training duration of the pipeline model, which is the sum of the training time of the two layers, is reported per fold by averaging over 10 folds (iterations: SSVM=80, AvGSPrec=20). The time tables 15 and 16 show that using the communicative inference for training global SpRL-SpQL is highly efficient. In fact the communicative inference converges in a few iterations, often fewer than 10 times. Therefore the efficiency of communicative training is comparable to the pipeline model.

Overall discussion. According to our experiments the results of the communicative inference during training are promising. Though the improvement compared to connecting the two independently trained global models for the two layers is small, it is consistently better and very efficient. The communicative learning will be effective if the two communicating models are sufficiently accurate (e.g. about 0.80 of F1 measure), otherwise the noisy feedback might drop the accuracy of the independent models. In our case the feedback of SpQL during training improves the model for the SpRL layer but not vice versa.

Pipe(SSVM)	Pipe(AvGSPerc)	Comm(AvGSPerc)
8h53m3s	1h16m9s	1h24m24s

Table 15. Training time per fold AvGSPerc (20 Iterations), SSVM (80 Itererations); averaged over 10 folds

Communicative	Pipeline
28.4s	15s

Table 16. Prediction time per fold; averaged over 10 folds

Label	Precision	Recall	F1
sp	0.769	0.701	0.733
tr	0.599	0.508	0.550
lm	0.489	0.396	0.438
r0	0.163	0.105	0.128

Table 17. SpRL layer, Confluence dataset, using training/evaluation of SemEval-2013, labeling headwords (LOSpRL).

Label	Precision	Recall	F1
sp	0.453	0.569	0.505
tr	0.395	0.711	0.508
lm	0.355	0.775	0.487
r0	0.146	0.202	0.170

Table 18. SpRL layer, Confluence dataset, using training/evaluation of SemEvam-2013, labeling headwords (LISpRL).

9.6. Extraction of Dynamic Spatial Relations

The goal is to investigate the influence of the global models and using constraints during training time and the prediction time in the LOSpRL, LISpRL and IBTSpRL models for the Confluence dataset. We use the constraints shown in formulas 7-8 and 9-12, which are the relevant ones for the SpRL layer annotations. Using only constraints 7-8 during training and prediction results in a simple LOSpRL model. The results of this setting are reported in Table 17. Except for the relation extraction, this setting provides reasonably good results particularly about recognition of the spatial indicators. In the second setting we added the constraints 9-12 during prediction to predict the roles and relations jointly (LISpRL). This setting improves the relation extraction sharply (5%) which is the most crucial element to be extracted correctly. However, providing more strict possible outputs decreases the recall of the spatial indicators dramatically. The extraction of the landmarks is improved in this setting about 5% compared to LOSpRL (see Table 18). In a more complex IBTSpRL setting we use the constraints during training as well as prediction. The results are reported in Table 19. This setting improves the recall of the spatial relations slightly (1%) and of the trajectors (3%). Since our model relies on the existence of the spatial indicators for the prediction of the roles, when there is no indicator in the sentence no role is predicted. To resolve this problem for the Confluence data which includes relations without indicators, in the last setting we allow *undefined* spatial indicators. In this way we obtain the results reported in Table 20 and achieve about 4% improvement in F1-measure for the relation extraction.

Overall discussion. The results of these experiments indicate the difficulty of correctly recognizing spatial relations and their composing spatial roles in the SpRL layer, which is mostly due to the lack of distinguishing linguistic features. The parse tree dependency path and the distance between words that are candidate trajectors, landmarks or spatial indicators are helpful but not very effective features in many cases. Providing distinguishing features for the relation extraction seems to be difficult using only the training data. In most cases, exploiting external resources and expansion/generalization over the lexical features can be helpful which is out of the scope of our study here. In this paper we observe that exploiting the constraints in a global and collective classification model can guide the models to discard the irrelevant relations when lacking distinguishing features in the training data. For example, when there is not enough evidence in the training data for classifying a word as a trajector but a preposition is recognized as spatial, then a constraint can help to impose the model to choose at least one word as a trajector in the sentence in its collective classification of IBT or LI settings. Such a collective setting then can improve the recall of the trajectors or even in some cases helps the model to realize that the preposition also should be classified as not spatial and improve the precision of the spatial indicator classification. In the IBT setting, by using the constraints during training when selecting the most violated output (y), we prohibit the training model from selecting irrelevant negative examples for updating parameters of the model (i.e. weights) and obtaining the optimal learning model.

9.7. Comparison with State-of-the-art Results

With regard to the recognition of the first layer of the ontology, SpRL, the literature reports the use of a conditional random fields [10], obtaining a precision, recall, and F1 of 0.79, 0.83 and 0.76 respectively, for the recognition of individual trajectors, 0.88, 0.92 and 0.84 for the recognition of individual landmarks, and 0.94, 0.92 and 0.96 for the recognition of individual spatial indicators. The link between the roles is indirectly implied by considering a spatial

Label	Precision	Recall	F1
sp	0.433	0.564	0.490
tr	0.389	0.744	0.511
lm	0.342	0.777	0.475
r0	0.141	0.210	0.169

Table 19. SpRL layer, Confluence dataset, using training/evaluation of SemEvam-2013, labeling headwords (IBTSpRL).

Label	Precision	Recall	F1
sp	0.473	0.709	0.567
tr	0.346	0.745	0.472
lm	0.299	0.691	0.417
r0	0.165	0.266	0.204

Table 20. SpRL layer, Confluence dataset, using training/evaluation of SemEvam-2013, labeling headwords allowing undefined spatial indicators (IBTSpRL).

pivot for each sequence during sequence tagging. In our work we directly recognize jointly the full spatial relations and their composing elements, with a precision, recall, and F1 of 0.67, 0.57 and 0.61 respectively, and consider the sentence level global constraints between all predicted relations. It is important to correctly recognize the spatial relation between a trajector and a landmark, instead of just knowing that a word functions as a trajector or landmark in the sentence.

In contrast to probabilistic graphical models such as conditional random fields, the proposed generalized linear model framework has the advantage of reduced computational complexity and the possibility to easily integrate global hard constraints, thus offering both efficiency and expressiveness. So far, we have not integrated sequential dependencies between labels of the words in a sentence as is done in the CRF, but we leave this to future work.

Bastianelli et al. [37] use an SVM-HMM classifier and a binary relation classifier in a pipeline where they first recognize the spatial indicator, trajector and landmark, and use this information to recognize the full spatial relations in the CLEF dataset. The results on the relation extraction are reported with F1= 0.358 for the strict evaluation and F1= 0.458 for the relaxed evaluation of the text spans. The results for the relation extraction seem to be lower than our results, however these are not fairly comparable since our results are on headwords and not on the phrases/ text spans. In the same paper the authors report precision, recall and F1 values of 0.60, 0.47 and 0.53 respectively, for spatial indicators, 0.56, 0.32 and 0.40 for trajectors, and 0.66, 0.47 and 0.56 for landmarks obtained on the Confluence data (SemEval-2013). These results are comparable to our results in table 20, though we report on (head) words again. It is important to notice that, there are no previous results reported on relation extraction on the Confluence data, which seems to be the most critical task. Bastianelli et al., for Confluence data, recognize spatial roles in isolation without recognizing the full relational structure, which we do in this paper. In their work a tree kernel is used, which provides a rich feature space and the lexical information are generalized via a distributional model of lexical semantics. We leave exploring other feature spaces to future work as this was not the focus of the work reported here.

With regard to the recognition of the second layer of the ontology, SpQL, or the joint recognition of the two layers or the full ontology, there are no available state-of-the-art studies.

10. Related Work

The ontology we use in this work is based on the spatial annotation scheme that we have proposed in a previous work [1]. We discuss the two layers of semantics and the adequacy of mapping to qualitative spatial representation and reasoning models in [12, 8, 9]. We have previously developed machine learning models, but they were restricted to the annotation of text with the concepts of the SpRL layer [10, 38]. The SpRL layer has been worked out by the participants of a semantic evaluation shared task that we have proposed [35, 39, 40, 37]. Preliminary classification of the ground truth spatial relations to RCC classes is investigated by the authors in [12] using an SMO implementation of the SVM multi-class classifier.

In this current paper, we extend the problem to cover an end-to-end mapping to both layers of the spatial ontology which is the first computational model for such a task. We choose structured prediction models which are able to model arbitrary structured output learning models.

From the ontology learning point of view our work is related to ontology population in which the textual instances of concepts and relations are assigned to the nodes of a predefined ontology. The related previous works in this area [41] mostly consider: a) an extensive preprocessing step applying NLP tools; b) external linguistic, web or relevant database resources; c) learning in pipeline models for extraction of the terms, concepts and the relationships

for classification or clustering over the extracted material; and d) postprocessing for resolving the inconsistencies in the predictions. According to a comprehensive study in [15] the related works are at the level of term, concept and relation extraction, and a few logic-based approaches discuss axiom extraction. To our knowledge there is no unified model proposed to extract the concepts defined by an ontology collectively in one unified framework of structured machine learning by considering the ontological relationships and background knowledge as global constraints, as we do in this work.

From the structured learning point of view the training is based on structured support vector machines and the structured perceptrons which are among the most well-known discriminative structured learning approaches and which perform well on different tasks such as question answering [42], natural language statistical parsing [24] and other domains [43]. Our applied structured learning formulation here is similar to the works in [26, 44, 23] and resembles a very generic formulation. There are other max-margin based formulations of structured output prediction which are problem specific or formulated for a certain type of loss functions [45, 46].

However, to be able to use the generic formulation of the structured learning in our context, we provide a novel relational formulation and component-based loss for the ontology population task. For the required inner inference we exploit combinatorial constraint optimization approaches. The structured model we propose is sufficiently general to be used for training ontologies that include a variety of ontological relationships. In previous work, usually very task-specific inference algorithms have been used for structured learning and those are designed for a specific structure of the output, for example a hierarchical structure as in [47].

From the relational learning point of view, given different components and types in our input and output we categorize our work as a kind of knowledge based model construction (KBMC) [48] that grounds the relational data and the background knowledge to a linear objective function along with linear constraints. The grounded objective is solved with combinatorial constraint optimization techniques in contrast to other KBMC techniques that compile the inference to a grounded probabilistic graphical model, for example as in Markov logic networks [49]. The most relevant relational learning models compared to our task, but applied on different domains and tasks, are link prediction [28] in web data and relation extraction from biomedical texts [29] in addition to natural language processing and information extraction models based on Markov logic networks [49, 50, 51]. All these works are based on learning and inference techniques in probabilistic graphical models. The advantage of our model is the use of more efficient optimization techniques and LP-relaxation for inference-based-training and also for prediction while exploiting global constraints. Using LP-relaxation techniques has found popularity in designing joint models for various natural language processing tasks [52]. To capture the global correlations in the probabilistic models, enough evidence in the data is required, which is difficult to obtain while using a small training dataset as we did. To specify our model we use the notion of templates as in relational graphical models to produce the objective function for each example. As in relational graphical models the clique templates [29] are grounded to produce the structure of the inference over each relational input, in our model the proposed grounded templates produce the monomials of a multilinear objective function which is linearized and solved subject to the linear constraints.

Exploiting global constraints in learning models during prediction is formally introduced in constrained conditional models [22]. Moreover compiling the propositional logical constraints for integer linear programming models is automated in a modeling language named learning based java (LBJ) [53]. In contrast to these works, by introducing a component based loss we are able to apply the same techniques for global training in addition to global prediction.

11. Conclusions

We have proposed a framework for representing the spatial semantics in natural language in terms of multiple calculi models. Moreover, a novel structured machine learning framework for mapping natural language to ontologies is provided. We propose a framework that we call Link-And-Label which is able to deal with relational data both in the input and in the output and is able to consider ontological relationships and background knowledge about the task during training and prediction. Using the notion of templates, we formalize the relational structure of learning in the inference-based training models. The objective function is produced by unrolling (i.e. grounding) the templates and producing a multinomial objective function to be optimized subject to the linearly grounded first order constraints. This formalization has been made for the first time and makes the usage of the combinatorial optimization for training and prediction in the presence of constraints straightforward. The proposed framework is applied to the novel problem of mapping natural language to spatial ontologies. Our results show the advantage of global learning compared to

learning independent classifiers and pipelining various parts of the ontology. Our communicative inference approach makes a global inference for the two layers of the spatial ontology comparatively efficient and accurate.

This work possesses a high potential for future extensions in its various dimensions. The constraints and the background knowledge from the ontology can be compiled automatically to their linear form from ontology languages such as OWL and directly used in the training and prediction models. This will be a very useful direction when working with ontologies used in the Semantic Web. The inference is also scalable and can be used to jointly recognize the concepts and their relationship defined in larger ontologies by decomposing them into smaller parts that can communicate to each other. In another direction, the spatial reasoning models can be integrated with our ontology population model and provide feedback to the learning models, for example, by spotting inconsistencies in the recognition of spatial information. The learned relations could be considered as probabilistic constraints about the most probable locations of the entities mentioned in a text and used in formal reasoning models that can deal with reasoning over uncertain spatial information. In addition, the spatial ontology itself can be extended to cover more fine-grained spatial notions, for example to cover dynamic spatial information and their formal representation, which implies the necessity of annotating data for training supervised models. The level of abstraction provided in our spatial ontology is amenable to and useful for many applications in different domains.

12. Acknowledgements

The research was funded by the KU Leuven grant DBOF/08/043, the EU FP7-296703 project MUSE (Machine Understanding for interactive Story tElling) and by the KU Leuven Postdoctoral grant PDMK/13/115.

References

[1] P. Kordjamshidi, M. van Otterlo, M. F. Moens, Spatial role labeling: task definition and annotation scheme, in: N. Calzolari, C. Khalid, M. Bente (Eds.), Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), 2010, pp. 413–420.

[2] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, E. Zavitsanos, Knowledge-driven multimedia information extraction and ontology evolution, Springer-Verlag, Berlin/Heidelberg, 2011, Ch. Ontology population and enrichment: state of the art, pp. 134–166. URL http://dl.acm.org/citation.cfm?id=2001069.2001075

[3] J. C. Bezdek, R. Hathaway, Some notes on alternating optimization, in: N. R. Pal, M. Sugeno (Eds.), Advances in Soft Computing, Vol. 2275 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002, pp. 288–300. doi:10.1007/3-540-45631-7-39.

[4] L. A. Carlson, S. R. Van Deman, The space in spatial language, Journal of Memory and Language 51 (2004) 418–436.

[5] J. Hois, O. Kutz, Natural language meets spatial calculi, in: C. Freksa, N. S. Newcombe, P. Gärdenfors, S. Wölfl (Eds.), Spatial Cognition VI. Learning, Reasoning, and Talking about Space, Vol. 5248 of LNCS, Springer, 2008, pp. 266–282.

[6] J. Renz, B. Nebel, Qualitative spatial reasoning using constraint calculi, in: M. Aiello, I. Pratt-Hartmann, J. van Benthem (Eds.), Handbook of Spatial Logics, Springer, 2007, pp. 161–215.

[7] J. A. Bateman, Language and space: A two-level semantic approach based on principles of ontological engineering, International Journal of Speech Technology 13 (1) (2010) 29–48.

[8] P. Kordjamshidi, M. van Otterlo, M. F. Moens, From language towards formal spatial calculi, in: R. J. Ross, J. Hois, J. Kelleher (Eds.), Workshop on Computational Models of Spatial Language Interpretation (CoSLI'10, at Spatial Cognition), 2010, pp. 17–24.

[9] P. Kordjamshidi, J. Hois, M. van Otterlo, M. F. Moens, Machine learning for interpretation of spatial natural language in terms of QSR, in: Extended abstract, 10th International Conference on Spatial Information Theory COSIT'11, 2011, pp. 1–5.

[10] P. Kordjamshidi, M. van Otterlo, M. F. Moens, Spatial role labeling: towards extraction of spatial relations from natural language, ACM - Transactions on Speech and Language Processing 8 (2011) 1–36.

[11] A. Galton, Spatial and temporal knowledge representation, Journal of Earth Science Informatics 2 (3) (2009) 169–187.

[12] P. Kordjamshidi, J. Hois, M. van Otterlo, M. F. Moens, Learning to interpret spatial natural language in terms of qualitative spatial relations, in: T. Tenbrink, J. Wiener, C. Claramunt (Eds.), Representing Space in Cognition: Interrelations of Behavior, Language, and Formal Models. Series Explorations in Language and Space, Oxford University Press, Oxford University Press, 2013, pp. 115–146.

[13] B. Kuipers, The spatial semantic hierarchy, Artificial Intelligence 119 (2000) 191–233.

[14] M. MacMahon, B. Stankiewicz, B. Kuipers, Walk the talk: Connecting language, knowledge, and action in route instructions, in: AAAI, 2006, pp. 1475–1482.

[15] W. Wong, W. Liu, M. Bennamoun, Ontology learning from text: a look back and into the future, ACM Computing Surveys 44 (4) (2012) 20:1–20:36.

[16] J. Zlatev, Holistic spatial semantics of Thai, Cognitive Linguistics and Non-Indo-European Languages (2003) 305–336.

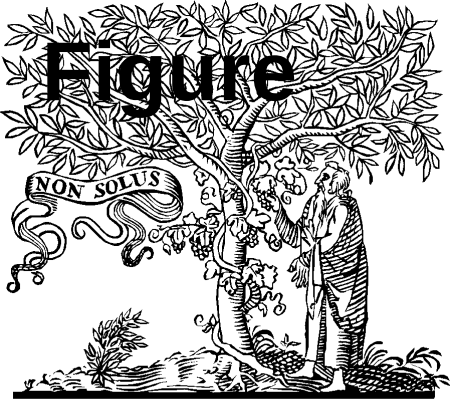
[17] J. Zlatev, Spatial semantics, in: D. Geeraerts, H. Cuyckens (Eds.), The Oxford Handbook of Cognitive Linguistics, Oxford Univ. Press, 2007, pp. 318–350.

[18] P. Kordjamshidi, Structured machine learning for mapping natural language to spatial ontologies, Ph.D. thesis, Katholieke Universiteit Leuven, Computer Science Department (2013).

[19] I. Mani, J. Pustejovsky, Interpreting Motion: Grounded Representations for Spatial Language, Explorations in language and space, Oxford University Press, 2012.
[20] D. A. Randell, Z. Cui, A. G. Cohn, A spatial logic based on regions and connection, in: Proceedings of the 3rd International Conference on the Principles of Knowledge Representation and Reasoning, KR'92, 1992, pp. 165–176.
[21] A. Klippel, R. Li, The endpoint hypothesis: a topological-cognitive assessment of geographic scale movement patterns, in: Spatial Information Theory, COSIT'09, 2009, pp. 177–194.
[22] M. W. Chang, L. A. Ratinov, D. Roth, Structured learning with constrained conditional models, Machine Learning 88 (3) (2012) 399–431.
[23] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, Journal of Machine Learning Research 6 (2) (2006) 1453–1484.
[24] M. Collins, Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, in: Proceedings of the Association for Computational Linguistics-02 Conference on Empirical Methods in Natural Language Processing, Vol. 10 of EMNLP '02, ACL, ACL, 2002, pp. 1–8.
[25] M. Collins, Parameter estimation for statistical parsing models: theory and practice of distribution-free methods, in: New Developments in Parsing Technology, Kluwer, 2004, pp. 19–55.
[26] B. Taskar, C. Guestrin, D. Koller, Max-margin Markov networks, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, MIT Press, 2004.
[27] B. Taskar, P. Abbeel, D. Koller, Discriminative probabilistic models for relational data, in: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, UAI'02, Morgan Kaufmann Publishers Inc., 2002, pp. 485–492.
[28] B. Taskar, M. F. Wong, P. Abbeel, D. Koller, Link prediction in relational data, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16, MIT Press, 2004.
[29] R. Bunescu, R. J. Mooney, Statistical relational learning for natural language information extraction, in: L. Getoor, B. Taskar (Eds.), Introduction to Statistical Relational Learning, MIT Press, 2007, pp. 535–552.
URL http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=51414
[30] J. Neville, D. Jensen, Relational dependency networks, Journal of Machine Learning Research 8 (2007) 653–692.
[31] D. P. Bertsekas., Nonlinear Programming, 2nd Edition, Athena Scientific, 1999.
[32] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of Optimization Theory and Applications 109 (2001) 475–494.
[33] V. Punyakanok, D. Roth, W. T. Yih, D. Zimak, Learning and inference over constrained output, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05, Morgan Kaufmann Publishers Inc., 2005, pp. 1124–1129.
[34] A. Vedaldi, A MATLAB wrapper of SVM^{struct}, http://www.vlfeat.org/ vedaldi/code/svm-struct-matlab.html (2011).
[35] P. Kordjamshidi, S. Bethard, M. F. Moens, SemEval-2012 task 3: Spatial role labeling, in: Proceedings of the First Joint Conference on Lexical and Computational Semantics: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval), Vol. 2, ACL, 2012, pp. 365–373.
[36] M. Grubinger, P. Clough, H. Müller, T. Deselaers, The IAPR benchmark: a new evaluation resource for visual information systems, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC), 2006, pp. 13–23.
[37] E. Bastianelli, D. Croce, R. Basili, D. Nardi, UNITOR-HMM-TK: Structured kernel-based learning for spatial role labeling, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 573–579.
URL http://www.aclweb.org/anthology/S13-2096
[38] P. Kordjamshidi, P. Frasconi, M. van Otterlo, M. F. Moens, L. De Raedt, Relational learning for spatial relation extraction from natural language, in: The Proceedings of ILP 2011, Lecture Notes in Artificial Intelligence, Vol. 7207, Springer, 2012, pp. 204–220.
[39] K. Roberts, S. Harabagiu, UTD-SpRL: a joint approach to spatial role labeling, in: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval'12), 2012, pp. 419–424.
[40] O. Kolomiyets, P. Kordjamshidi, M. F. Moens, S. Bethard, Semeval-2013 task 3: Spatial role labeling, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 255–262.
URL http://www.aclweb.org/anthology/S13-2044
[41] J. M. Ruiz-Martìnez, J. A. Minarro-Gimènez, D. Castellanos-Nieves, F. Garcìa-Sànchez, R. Valencia-Garcia, Ontology population: an application for the E-tourism domain, International Journal of Innovative Computing, Information and Control (IJICIC) 7 (11) (2011) 6115–6134.
[42] Y. Cao, W. Y. Yang, C. Y. Lin, Y. Yu, A structural support vector method for extracting contexts and answers of questions from online forums, Information Processing and Management 47 (6) (2011) 886–898.
[43] S. Nowozin, C. H. Lampert, Structured learning and prediction in computer vision, Foundations and Trends in Computer Graphics and Vision 6 (3-4) (2011) 185–365.
[44] R. Samdani, D. Roth, Efficient decomposed learning for structured prediction, in: Proceedings of the 29th International Conference on Machine Learning, 2012.
[45] C. H. Lampert, Maximum margin multi-label structured prediction, in: J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 24, 2011, pp. 289–297.
[46] X. Qiu, W. Gao, X. Huang, Hierarchical multi-class text categorization with global margin maximization, in: Proceedings of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing, 2009, pp. 165–168.
[47] Y. Li, K. Bontcheva, Hierarchical, perceptron-like learning for ontology-based information extraction, in: Proceedings of the 16th international conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 777–786. doi:10.1145/1242572.1242677.
[48] M. P. Wellman, J. S. Breese, R. P. Goldman, From knowledge bases to decision models, Knowledge Engineering Review 7 (1) (1992) 35–53.
[49] P. Domingos, M. Richardson, Markov logic: A unifying framework for statistical relational learning, in: ICML'04 Workshop on Statistical Relational Learning and its Connections to Other Fields, 2004, pp. 49–54.

[50] H. Poon, P. Domingos, Joint inference in information extraction, in: Proceedings of the 22nd National Conference on Artificial Intelligence, Vol. 1 of AAAI'07, 2007, pp. 913–918.
[51] I. Meza-Ruiz, S. Riedel, Jointly identifying predicates, arguments and senses using Markov logic, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL, ACL, 2009, pp. 155–163.
[52] A. M. Rush, M. Collins, A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing., The Journal of Artificial Intelligence Research (JAIR) 45 (2012) 305–362.
[53] N. Rizzolo, D. Roth, Learning based Java for rapid development of NLP systems, in: Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010.

Figure



ELSEVIER