Exploiting Transitivity in Probabilistic Models for Ontology Learning

Francesca Fallucchi[1] and Fabio Massimo Zanzotto[2]
[1]University of Rome "G. Marconi", Italy [2] University of Rome "Tor Vergata", Italy

ABSTRACT

Capturing word meaning is one of the challenges of natural language processing (NLP). Formal models of meaning such as ontologies are knowledge repositories used in a variety of applications. To be effectively used, these ontologies have to be large or, at least, adapted to specific domains. Our main goal is to contribute practically to the research on ontology learning models by covering different aspects of the task.

We propose probabilistic models for learning ontologies that expands existing ontologies taking into accounts both corpus-extracted evidences and structure of the generated ontologies. The model exploits structural properties of target relations such as transitivity during learning. We then propose two extensions of our probabilistic models: a model for learning from a generic domain that can be exploited to extract new information in a specific domain and an incremental ontology learning system that put human validations in the learning loop. This latter provides a graphical user interface and a human-computer interaction workflow supporting the incremental leaning loop.

INTRODUCTION

Gottfried Wilhelm Leibniz was convinced that human knowledge was like a *"bazaar"*: a place full of all sorts of goods without any order or inventory. As in a *"bazaar"*, searching a little piece of specific knowledge is a challenge that can last forever. Nowadays, we have powerful machines to process and collect data. These machines, combined with the human need of exchanging and sharing information, produced an incredibly large evolving collection of documents, partially shared with the World Wide Web. The Web is a modern worldwide scale knowledge *"bazaar"* full of any sort of information where searching specific information is a titanic task.

Ontologies represent the Semantic Web's reply to the need of searching knowledge in the Web. These ontologies provide shared metadata vocabularies (Berners-Lee, T., Hendler, J., & Lassila, O., 2001). Data, documents, images, and information sources in general, described through these vocabularies, will be thus accessible as organized with explicit semantic references for humans as well as for machines. Yet, to be useful, ontologies should cover large part of human knowledge. Automatically learning these ontologies from document collections is *the major challenge*.

Models for automatically learning semantic networks of words from texts use both corpus-extracted evidences and existing language resources (Basili, Gliozzo, & Pennacchiotti, 2007). All these models rely on two hypotheses: *Distributional Hypothesis* (*DH*) (Harris, 1964) and *Lexico-syntactic patterns exploitation hypothesis* (LSP) (Robison, 1970). While these are powerful tools to extract relations among concepts using texts, models based on these hypotheses do not explicitly exploit structural properties of target relations when learning taxonomies or semantic networks of words. DH models intrinsically use structural properties of semantic networks of words such as transitivity, but these models cannot be applied for learning transitive semantic relations other than the generalization. LSP models are interesting because they can learn any kind of semantic relations. Yet, these models do not exploit structural properties of target relations when learning taxonomies or semantic networks of words. In general, structural properties of semantic networks of words, when relevant, are not used in machine learning models to better induce confidence values for extracted semantic relations. Even where transitivity is explicitly used (Snow, Jurafsky, & Ng, 2006), it is not directly exploited to model confidence values. It is only used in an iterative maximization process of the probability of the entire semantic network. In this chapter, we propose a probabilistic approach that exploits LSP hypothesis and formally includes the exploitation of transitivity during learning.

Probabilistic models for learning semantic networks exploiting transitivity do not completely solve the problem of learning semantic networks. We have a second problem to tackle. When dealing with learning semantic networks of words from texts such as learning ontologies, we generally have *ontology-rich* domains with large structured domain knowledge repositories or large general corpora with large general structured knowledge repositories such as WordNet (Miller, 1995). Systems that automatically create, adapt, or extend existing semantic networks of words need a sufficiently large number of documents and existing structured knowledge to achieve reasonable performance. Thus, it is generally possible to extract good probabilistic models for *ontology-rich* domains or the general language. When building semantic networks for *ontology-poor* domains, we then need to rely on probabilistic models learnt out-of-domain or for the general language. If the target domain has not relevant pre-existing semantic networks of words to expand, we will not have enough data for training the initial model. In general, in learning methods the amount of out-of-domain data is larger than in-domain data. For this reason, in this chapter we present methods that, with a small effort for the adaptation to different specific knowledge domains, can exploit out-of-domain data for building in-domain models with bigger accuracy.

Finally, when learning semantic networks, we need to put human validations in the loop. Systems for creating or augmenting semantic networks of words using information extracted from texts need a manual validation for assessing the quality of semantic networks of words expansion. Yet, these systems do not use the manual validation for refining the information extraction model that proposes novel links in the networks. Manual validation can be efficiently exploited if used in an incremental model. In this chapter, we propose an incremental ontology learning system that puts final users in the learning loop providing an efficient way to interact with final users.

The rest of the Chapter is organized as follows. In Section **Methods for Ontology Learning** we give a survey of the main strategies and approaches, nowadays adopted, in learning semantic networks of words. In particular, we propose a review of the state-of-the-art and we point out the limits that can be overcome with our approaches. In Section **Transitivity in a Probabilistic Model** we introduce our probabilistic models to learn semantic networks of words that exploit structural properties of target relations in determining the probability of the word pairs to be in a particular relation. Then we present two extensions of our probabilistic model: a semantic networks learning method that can exploit models learned from a generic domain to extract new information in a specific domain, in Section **Generic Ontology Learners on Application domains**, and an incremental ontology learning system that puts final users in the learning loop and uses our probabilistic models to exploit transitive relations for inducing better extraction models, in Section **Probabilistic Ontology Learner in Semantic Turkey**. Finally, we draw some conclusions and we outline feature research directions.

METHODS FOR ONTOLOGY LEARNING

Automatically creating, adapting, or extending existing ontologies or semantic networks of words using domain texts is a very important and active area of research. Here, we report the state-of-the-art of learning semantic networks of words, that is the field where this chapter wants to give a contribution.

Ontology learning was originally started in (Maedche & Staab, Ontology Learning for the Semantic Web, 2001) but the fully automatic acquisition of knowledge by machines is still far from being realized. Ontology learning is not merely a rehash of existing ideas and techniques under a new name. Lexical acquisition, information extraction, knowledge base learning from texts, etc. are areas that contribute to the definition of this new problem. But, ontology learning is more than the sum of all these contributions. This new problem is inherently multidisciplinary due to its strong connection with philosophy, knowledge representation, database theories, formal logic, and natural language processing. Moreover, as ontologies are the basis for the Semantic Web, learning models have to work with massive and heterogeneous data and document collections.

In natural language processing and in many applications of the semantic web, semantic resources are ultimately exploited in text understanding systems as networks of words. Thus, learning semantic networks from text collections is possible. Here we focus on the learning of relations among concepts/words. In the following we analyze these techniques thoroughly, with particular reference to aspects and components that characterize them, limitations included. We focus on the three aspects the chapter deals with: a general introduction on semantic network learning methods to present the limitations with respect to the use of structural properties of target semantic relations; a general discussion of the problem of domain adaptation; and, finally, an analysis of the methods to include human validations in the learning loop.

Semantic network learning methods

Models for automatically learning semantic networks of words from texts use both corpus-extracted evidences and existing language resources (Basili, Gliozzo, & Pennacchiotti, 2007). All these models rely on two hypotheses: *Distributional Hypothesis* (*DH*) (Harris, 1964) and *Lexico-syntactic patterns exploitation hypothesis* (LSP) (Robison, 1970). In this section we focus on how existing resources are used in the existing learning models and we note that these models do not explicitly exploit structural properties of target relations when learning taxonomies or semantic networks of words.

Distributional Hypothesis (*DH*) (Harris, 1964) models generally start learning from scratch. In (Cimiano, Hotho, & Staab, Learning concept hierarchies from text corpora using formal concept analysis, 2005), for example, lattices and related semantic networks are built from scratch. When prior knowledge is used in DH models (Pekar & Staab, 2002), the status of prior knowledge and of produced knowledge is extremely different. Inserting new words in semantic networks may be seen as a classification problem where target classes are nodes of existing hierarchies and the classification decision is taken over a pair of words, i.e. a word and its possible generalization. In this context, the classifier should decide if pairs belong or not to the semantic networks. Both existing and produced elements of the networks have the same nature, i.e., pairs of words. A distributional description of words is used to make the decision with respect to target classes. A new word and a word already existing in the network can be then treated differently, the first being represented with its distributional vector while the second being one of the final classes.

DH is widely used in many approaches in relation induction from texts. Relatedness confidences derived using the distributional hypothesis are transitive. If a word "*a*" is related to a word "*b*" and this latter is related to a word "*c*", we can somehow derive the confidence relations between the words "*a*" and "*c*". This can be derived from the formulation of the distributional hypothesis itself. Even when the distributional hypothesis is used to build hierarchies of words, structural properties of the semantic networks of words, such as transitivity and reflexivity are implicitly used. For example, DH is used in (Cimiano, Hotho, & Staab, Learning concept hierarchies from text corpora using formal concept analysis, 2005) for populating lattices (i.e. graphs of a particular class) of formal concepts. The idea of drawing semantic networks links using the inclusion of features derived exploiting the distributional hypothesis has been also used in (Geffet & Dagan, 2005) where the *distributional inclusion hypothesis* is defined.

Lexico-syntactic patterns (LSP) (Robison, 1970) are instead generic ways to express a semantic relation in texts.-LSP models have been applied for learning is-a relations (Hearst, 1992; Snow, Jurafsky, & Ng, 2006), generic semantic relations between nouns (Pantel & Pennacchiotti, 2006; Szpektor, Tanev, Dagan, & Coppola, 2004), and specific relations between verbs (Chklovski & Pantel, 2004; Zanzotto, Pennacchiotti, & Pazienza, Discovering Asymmetric Entailment Relations between Verbs Using Selectional Preferences, 2006). But, LSP models do not directly exploit structural properties of semantic networks of words, i.e. these properties are not intrinsically inherited from the definition, as it differently happens for the distributional hypothesis. Semantic network learning models based on lexico-syntactic patterns present then three advantages with respect to DH models:

- these models can be used to learn any semantic relation (Hearst, 1992; Morin, 1999; Pantel & Pennacchiotti, 2006; Chklovski & Pantel, 2004; Ravichandran & Hovy, 2002; Szpektor, Tanev, Dagan, & Coppola, 2004; Zanzotto, Pennacchiotti, & Pazienza, Discovering Asymmetric Entailment Relations between Verbs Using Selectional Preferences, 2006)

- these models coherently exploit existing taxonomies in the expansion phase (Snow, Jurafsky, & Ng, 2006)
- the classification is binary, i.e., a word pair belongs or not to the taxonomy (Snow, Jurafsky, & Ng, 2006; Pantel & Pennacchiotti, 2006). In this way, a single classifier is associated to each treated relation.

Probabilistic LSP learning models (e.g., (Snow, Jurafsky, & Ng, 2006)) have further advantages. The first advantage is that modeling probability in the semantic network makes possible to take into accounts both corpus-extracted evidences and existing language resources during learning. This model is the only one using, even if only intrinsically, one of the properties of semantic networks (transitivity) to expand existing networks. Any corpus-based knowledge learning method augments existing knowledge repositories with new information extracted from texts. In this process, we have two big issues:

- we are mixing reliable with unreliable information
- as we are dealing with natural language, ambiguity affects every bit of discovered knowledge

Mixing reliable concepts, relations among concepts, and instances with semi-reliable extracted information is a big problem as final knowledge repositories cannot be considered reliable. Generally, extracted knowledge items are included in final resources if the related estimated confidence weights are above a threshold. Accuracy of added information is generally evaluated over a small randomly selected portion (e.g., (Snow, Jurafsky, & Ng, 2006; Pantel & Pennacchiotti, 2006; Lin & Pantel, 2002)). Final knowledge repositories contain, then, two different kinds of information. The first kind is reliable and controlled. The second kind, i.e., the above threshold extracted information, is semi-reliable. Its accuracy is below 100% and it generally varies in different ranges of confidence weights. High confidence values guarantee high accuracy (e.g., (Snow, Jurafsky, & Ng, 2006)). Then, it is extremely important that corpus extracted knowledge items report the confidence weights that justifies the inclusion in the knowledge base. In this way, *consumers* of knowledge repositories can decide if information is “reliable enough” to be applied in their task. This is the first reason to include probability scores in knowledge repositories.

The second advantage is that stored probabilities enable the treatment of the ambiguity of natural language. For example, the word “*dog*” can be generalized to the word “*animal*” or to the word “*device*” according to which sense is taken into account. A decision system working with words would benefit in accuracy from the knowledge of the probabilities of two different generalizations. The simple ordering of word senses in WordNet (Miller, 1995) (first sense heuristic) according to their frequencies is useful for open domain word sense disambiguation models. Also the computation of prior sense probabilities within specific domains is useful for word sense disambiguation processors (McCarthy, Koeling, Weeds, & Carroll, 2004).

We will select a probabilistic approach, among LSP semantic networks learning models, because in this way we can have the two described advantages.

Adapting semantic networks to new domains

In learning methods the amount of *out-of-domain* data is generally larger than *in-domain* data. For this reason, we envisage methods that, with a small effort for the adaptation to different specific knowledge domains, can exploit *out-of-domain* data for building *in-domain* models with bigger accuracy. We would like a model for learning semantic networks of words that can be used, with a small effort for the adaptation, in different specific knowledge domains.

One of the basic assumptions in machine learning and statistical learning is that learning data are enough representative of the environment where learned models will be applied. The statistical distribution of learning data is similar to the distribution of the data where the learned model is applied. In natural language processing tasks involving semantics, this assumption is extremely important. One of these semantic tasks is learning semantic networks of words from texts using lexico-syntactic pattern (LSP) based methods. LSP methods (Hearst, 1992; Snow, Jurafsky, & Ng, 2006; Pantel & Pennacchiotti, 2006) generally use existing ontological resources to extract learning examples. The learning examples are matched over collection of documents to derive lexico-syntactic patterns describing a semantic relation. These patterns are then used to expand the existing ontological resource by retrieving and selecting new examples. LSP semantic networks learning methods are generally used to expand existing domain ontologies using domain corpora or to expand generic lexical resources (e.g.,WordNet (Miller, 1995)) using general corpora (Snow, Jurafsky, & Ng, 2006; Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009) (Snow, Jurafsky, & Ng, 2006; Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009). In this way, the basic assumption of machine learning approaches is satisfied. Yet, the nature of the semantic networks learning task requires that models learned in a general or a specific domain may be applied in other domains for building or expanding poor initial semantic networks using domain corpora. In this case, the distribution of learning and application data is different. Learned LSP models are "domain-specific" and they being potentially related to the prose of a specific domain. These models are then accurate for the specific domain but may fail in other domains. For examples, if the target domain has not relevant pre-existing ontologies to expand, may be not enough data for training the initial model. In (Snow, Jurafsky, & Ng, 2006), all WordNet has been used as source of training examples. In this case, domain adaptation techniques must be adopted (Bacchiani, Roark, & Saraclar, 2004; Roark & Bacchiani, 2003; Chelba & Acero, 2006; Gao, 2009; Gildea, 2001).

Domain adaptation is a well-known problem in machine learning and statistical learning. The problem of domain adaptation arises in a large variety of applications: natural language processing (Chelba & Acero, 2006; Blitzer, Mcdonald, & Pereira, 2006), machine translation (Bertoldi & Federico, 2009), word sense disambiguation (Chan & Ng, 2007), etc...

Different domain adaptation techniques are introduced in the context of specific applications and statistical learning methods. One of the possible ways of using the model adaptation is to adjust the model trained on the background domain to a different domain (the adaptation domain) modifying opportunely the parameters and/or the structure. The motivation of this approach is that usually the background domain has large amounts of training data while the adaptation domain has only small amounts of data. By analogy with (Blitzer, Mcdonald, & Pereira, 2006) we propose to learn common features, meaningful for both domains having different weights, where the weights are determined according to the occurrences in the respective corpus. We are confident that a model trained in the source domain using this common feature representation will generalize better the target domain.

Systems for creating or augmenting semantic networks of words using information extracted from texts foresee a manual validation for assessing the quality of semantic networks of words expansion. Yet, these systems do not use the manual validation for refining the information extraction model that proposes novel links in the networks. Manual validation can be efficiently exploited if used in an incremental model. We need an efficient way to interact with final users.

Incremental Ontology Learning

Exploiting the above (and also other) algorithms and techniques for inducing ontological structures from texts, different approaches have been devised, followed and applied regarding how to properly exploit the learned objects and how to translate them into real ontologies using dedicated editing tools. This is an aspect which is not trivially confined to importing induced data inside an existing (or empty) semantic network, but identifies iterative processes that could benefit from properly assessed interaction steps with the user, giving life to novel ways of interpreting semantic networks development.

One of the most notable examples of integration between semantic networks learning systems and ontology development frameworks is offered by Text-to-Onto (Maedche & Volz, ICDM Workshop on integrating data mining and knowledge management, 2001), an ontology learning module for the KAON tool suite, which discovers conceptual structures from different kind of sources (ranging from free texts to semi-structured information sources such as dictionaries, legacy ontologies and databases) using knowledge acquisition and machine learning techniques; OntoLT (Buitelaar, Olejnik, & Sintek, 2004) is a Protégé (Gennari, Musen, Fergerson, Grosso, Crubzy, & Eriksson, 2003) plug-in able to extract concepts (classes) and relations (Protégé slots or Protégé OWL properties) from linguistically annotated text collections. It provides mapping rules, defined by use of a precondition language, that allow for a mapping between extracted linguistic entities and classes/slots.

An outdated overview of this kind of integrated tools (which is part of a complete survey on ontology learning methods and techniques) can be found in the public Deliverable 1.5 (Gómez-Pérez & Manzano-Macho, 2003) of the OntoWeb project. A more recent example is offered by the Text2Onto (Cimiano & Volker, Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery, 2005) plug-in for the Neon toolkit (Haase, Lewen, Studer, Tran, d'Aquin, & Motta, April, 2008), a renewed version of Text-To-Onto with improvements featuring on-model independence (a *Probabilistic Ontology Model* is adopted as a replacement for any definite target ontology language), better user interaction and incremental learning.

Lastly, in (Bagni, Cappella, Pazienza, Pennacchiotti, & Stellato, 2007) the authors define a web browser extension based on the Semantic Turkey Knowledge Acquisition Framework (Griesi, Pazienza, & Stellato, Semantic Turkey - a Semantic Bookmarking tool (System Description), 2007), offering two distinct learning modules: a relation extractor based on a light-weight and fast-to-perform version of algorithms for relation extraction defined in (Pantel & Pennacchiotti, 2006), and an ontology population module for harvesting data from html tables. Most of the above models defines supervised cyclic *develop and refine* processes controlled by domain experts. We propose to extend Semantic Turkey (ST) integrating ST with our novel probabilistic model to put final users in the learning loop with an efficient way to interact with final users.

In the rest of the chapter we propose solutions to some limits seen in this section. In particular we propose models to exploit structural properties of target relations such as transitivity during learning process. Then, we introduce two applications that use our probabilistic model: a model that can be used in different specific knowledge domains with a small effort for its adaptation and a model that allows to put final users in the learning loop for adapting the model.

TRANSITIVITY IN A PROBABILISTIC MODEL

Capturing word meaning is one of the challenges of natural language processing. Taxonomies and, in general, semantic networks of words (Miller, 1995) are often used as formal models of word meaning. In these networks, words are connected with other words by means of taxonomic and, in general, semantic relations. This is a way to capture part of the knowledge described in traditional dictionaries. For example, this informal definition of "*wheel*":

a ***wheel*** *is a circular frame turning about an axis ... used for supporting vehicles...*

contains a *taxonomic relation*, i.e., *the wheel is a circular frame*, and a sort of *part-of relation*, i.e., *the wheel is used for supporting vehicles*.

Transitivity is a well known property of some foundational semantic relations between words. Semantic networks are built over transitive semantic relations such as generalization, cotopy, meronymy, cause-effect, entailment, and so on. Knowing that "*dog*" is a "*mammal*" and "*mammal*" is a "*animal*", we can infer that "*dog*" is a "*animal*" or, knowing that "*snoring*" *e*ntails "*sleeping*" and "*sleeping*" *e*ntails "*resting*", we can state that "*snoring*" *e*ntails "*resting*". Yet, this property is generally not exploited in learning semantic relations from texts.

The semantic networks learning models do not explicitly exploit properties, such as transitivity, when learning taxonomies or networks of words. Transitivity, when relevant, is not used to better induce confidence values for extracted semantic relations. Even where transitivity is intrinsically used (Snow, Jurafsky, & Ng, 2006), it is not directly exploited to model confidence values but it is used in an iterative maximization process of the probability of the entire semantic network. We transform this limitation into an opportunity. In particular we propose a novel probabilistic method for learning semantic networks of words that explicitly models transitivity for deriving confidence weights.

The rest of the section is organized as follows. We informally introduce our probabilistic model that explicitly used transitivity in semantic networks learning models. Then, we formalize the probabilistic definitions of concepts in an *induced* probabilistic model and we propose three different methods for modeling induced probabilities. Finally, we want to demonstrate that our *induced* models can effectively exploit transitivity when we replicate existing networks or we expand or build new semantic networks.

Probabilistic definitions of concepts in semantic networks learning

When we consider semantic relations with structural properties as transitivity, including confidence weights in knowledge repositories is not a trivial problem. In methods such as (Pantel & Pennacchiotti, 2006), it seems to be possible to easily include some initial values in the final resource as these have been used for deciding whether or not the knowledge base should include a relation. Yet, when we need to combine these values in transitive relations, we need to be extremely careful on how these values have been estimated and computed. For example, if we discover from corpus analysis that "*dog*" is a "*canine*" and we already know that "*canine*" is an "*animal*" (see Figure 1(a)), using transitivity we can derive the *induced* relation, i.e., *dog* is an *animal* (dashed arrow in Figure 1(a)). Yet, we cannot easily combine confidence weights if the nature of these weights is obscure. On the contrary if we discover from corpus analysis that "*dog*" is an "*animal*" and we already know that "*dog*" is "*canine*" (see Figure 1(b)), using the transitivity we can derive the *induced* relation, i.e., *canine* is an *animal* (dashed arrow in Figure 1(b)). Another example is shown in Figure 1(c). The solution generally proposed for combining confidence weights is neglecting its nature. The final relation between two words has the same confidence weight of reliable and controlled information.

Even in the probabilistic models (Snow, Jurafsky, & Ng, 2006), these reliable and unreliable information is mixed during the knowledge acquisition process. In these models, if "*canine*" is an "*animal*" (see Figure 1(a)) is in the original manually controlled network and "*dog*" is a "*canine*" has a high probability from the corpus observations, this latter is included in the knowledge base with the same degree of plausibility of "*canine*" is an "*animal*". Then, the induced relation "*dog*" is an "*animal*" has again the same degree of plausibility of manually controlled information. This represent a loss of information the uncertainty of the relation "*dog*" is an "*animal*" has been neglected.

Figure 1: Examples of relations derived exploiting the transitivity

Probabilistic definitions for concepts

Keeping and propagating uncertainty in transitive semantic networks is extremely important. We thus propose an *inductive semantic network learning model*, i.e., a probabilistic semantic network learning model based on lexico-syntactic patterns that exploits transitivity during learning and for determining combined confidence weights. Our model stems from the intuition that LSP learning models contribute to *probabilistic definitions of target concepts* and that it is possible to combine these definitions to determine confidence weights derived from the transitive networks. Extracting evidence from corpora suggesting that "*dog*" is an "*animal*" contributes both to the definition of "*dog*" and to the definition of "*animal*". In the case of "*dog*", the relation between "*dog*" and "*animal*" contributes to the intensional definition of "*dog*", it stating that "*dog*" is an "*animal*" with specific features. In the case of "*animal*", this relation contributes, in a wide sense, to the *extensional* definition[i] of "*animal*". It is like we are giving one of the possible instances[ii] of the concept "*animal*". These formal *intensional* and *extensional* definitions are often used to derive the similarity among words or concepts. *Cotopy* (Maedche & Staab, Measuring

Similarity between Ontologies, 2002), a measure for determining similarity between concepts in two different semantic networks, uses exactly this information.

A *probabilistic definition* of a concept is an intensional definition associated with its *induced* probabilities. These probabilities are derived from the topology of the transitive semantic networks mixing existing knowledge and corpus estimated probabilities. In Figure 1, the solid arrow indicates relations derived from existing structured knowledge repositories and from corpus analysis while the dashed arrow type indicates probabilities induced from the structure of the network. We want to describe the probability of the dashed relations using the probabilities of the solid ones. We call *direct probabilities* the first type, and *induced probabilities* the second one.

In accord to (Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009), we define the *direct probabilities* as the direct events $R_{i,j} \in T$ where T is the semantic network. If $R_{i,j}$ is in T , then "i" is in a R relation with "j" according to the semantic network T. For example, if R is the is-a relation, $R_{dog,animal} \in T$ describes that dog is an animal according to the semantic network T . The learning problem in the direct settings is to determine the probabilities:

$$P\ (R_{i,j}\ \in\ T\ |E)$$

Starting from the idea described above, we propose three models that derive *induced probabilistic definitions* from *direct probabilities*: the first exploits *intensional* definitions of concepts while the second exploits *extensional* definitions and the third exploits both *intensional* and *extensional* probabilistic definitions of concepts. We then define the three models respectively: the *intensional*, the *extensional* and the *mixed probabilistic inductive model*. To give an intuitive idea of our models, we can use the example in Figure 1.

The *intensional inductive model* exploits direct *intensional* definitions to derive an induced *intensional* definition. In Figure 1(a), we have as direct information the probabilities of the relations "*dog*" is a "*canine*" and "*canine*" is a "*animal*". From these two relations, we can derive the induced probability of the intensional definition of "*dog*" is a "*animal*". In this case we are exploiting and modeling the transitivity of the isa relation.

The *extensional inductive model* uses the direct probabilities (solid arrows), to form *extensional* definitions of the concepts and, to compare the different *extensional* definitions for determining the final induced probability. In Figure 1(b), the relations "*dog*" is a "*animal*" and "*dog*" is a "*canine*" are used to form a very small part of the *extensional* definitions of, respectively, "*animal*" and "*canine*". The idea is that these *extensional* definitions can be used to determine the similarity of "*animal*" and "*canine*". Then, we can derive the induced probability of the relation "*dog*" is a "*animal*". Using the same intuition, the relations "*dog*" is a "*animal*" and "*canine*" is a "*animal*" contribute to the *extensional* definition of "*animal*" (see Figure 1(c)). Using all the other relations, we can derive also the induced probability of the relation "*dog*" is a "*canine*".

Inductive Probabilistic Model

In this section, we formalize the probabilistic definitions of concepts in an *induced* probabilistic model. We introduce three models for exploiting the probabilistic definitions of concepts within the *induced* probabilistic model. Without loss of generality, we focus the examples and the prose on semantic networks learning. Yet, these models can be adopted for any transitive semantic relation.

As in (Pantel & Pennacchiotti, 2006; Snow, Jurafsky, & Ng, 2006), we model the semantic networks learning problem as a binary classification task. Given a pair of words (i,j) and a vector of observed features $\overrightarrow{e_{i,j}}$, we want to build a binary classifier that determines if i is a j and gives the related confidence weight. As in (Snow, Jurafsky, & Ng, 2006), we see this problem in a probabilistic point of view as it gives the possibility to determine the *direct probabilistic model* as well as the *induced probabilistic model.*

Figure 2: Example of relations derived exploiting transitivity

We here propose a model to exploit transitivity within probabilistic semantic networks learners that use lexico-syntactic patterns. Using lexico-syntactic patterns on a corpus, we can extract pairs of words in a given relation along with their reliability. These pairs of words and their reliabilities are *directly* observed. For example (see Figure 2), given the hyperonymy relation, we *directly* derive the reliabilities of the pairs "*dog*" is a "*canine*" (0.8), "*canine*" is an "*animal*" (0.7), and "*dog*" is an "*animal*" (0.2) (solid arrows). If we now look at all these pairs as a whole, we can observe that these words form a semantic network where transitive property holds. Even if the *directly* observed reliability of the pair "*dog*" is an "*animal*" is low (0.2), transitivity of the network suggests that this reliability should be higher (0.648). We exactly want to exploit the transitive network to *induce* the reliability of the relation between "*dog*" and "*animal*" (dashed arrow) using all the reliabilities of the involved pairs *directly* observed from the corpus. We then use a probabilistic setting where this composition of confidence weights can be better controlled.

The example of Figure 2 we have the following *direct* probabilities (where *d=dog*, *a=animal*, and *c=canine*): $P\left(R_{d,a}\middle|\vec{e}_{d,a}\right) = 0.2$, $P\left(R_{d,c}\middle|\vec{e}_{d,c}\right) = 0.8$ and $P\left(R_{c,a}\middle|\vec{e}_{c,a}\right) = 0.7$.

In the *inductive probabilistic model* presents the main innovation of our approach to semantic networks learning. We want here to define an event space that models transitivity. We then introduce the events $\hat{R}_{i,j}$ and the related probability function:

$$P(\hat{R}_{i,j} \in T|E) \qquad (1)$$

This probability should capture the fact that a decision on the pair (i,j) also depends on the transitive relations activated by (i,j). Rarely these relations are activated by *existing semantic networks* links. Yet, this *induced* probability takes into account transitively related taxonomic links. We examine different models to exploit the transitive property of the R relation and for each of these models we show that $P(\hat{R}_{i,j}|E)$ can be rewritten in term of the involved $P(R_{h,k}|E)$.

For example, we can compute the *induced intensional* probability for the pair (*dog*,*animal*) in Figure 2. The *induced intensional* probability $P(\hat{R}_{d,a}|E)$ can be computed as the probability of the event $\hat{R}_{d,a} = R_{d,a} \cup (R_{d,c} \cap R_{c,a})$. This captures that the *induced* event $\hat{R}_{d,a}$ is active when $R_{d,a}$ happens or the joint event $R_{d,c} \cap R_{c,a}$ happens. Then, using the inclusion-exclusion property, the previous independence assumptions on the evidences E, and an independence assumption between $R_{i,j}$, we can compute $P(R_{d,a} \cup (R_{d,c} \cap R_{c,a})|E)$ as:

$$\begin{gathered} P\,(R_{d,a}\,\cup\,(R_{d,c}\,\cap\,R_{c,a}\,)|E)\,= \\ =\,P\,(R_{d,a}|E)\,+\,P\,(R_{d,c}\,\cap\,R_{c,a}|E)\,-\,P\,(R_{d,a}\,\cap\,R_{d,c}\,\cap\,R_{c,a}\,|E)\,= \\ =\,P\,(R_{d,a}|\vec{e}_{d,a}\,)\,+\,P\,(R_{d,c}\,|\vec{e}_{d,c}\,)P\,(R_{c,a}\,|\vec{e}_{c,a}\,)\,-\,P\,(R_{d,a}\,|\vec{e}_{d,a}\,)P\,(R_{d,c}\,|\vec{e}_{d,c}\,)P\,(R_{c,a}\,|\vec{e}_{c,a}\,)\,= \\ =\,0.2\,+\,0.8\,*\,0.7\,-\,0.2\,*\,0.8\,*\,0.7\,=\,0.648 \end{gathered}$$

We propose three different methods for modeling induced probabilities: *intensional*, *extensional*, and *mixed* model as described in (Fallucchi & Zanzotto, Inductive Probabilistic Taxonomy Learning using Singular Value Decomposition, 2010). These three models exploit different definitions of the event $\hat{R}_{i,j} \in T$.

In the *intensional* model, the event $R_{i,j} \in T$ is represented as the event $R_{i,j} \in T$ and for any k all the alternative events $R_{i,k} \in T$ and $R_{k,j} \in T$. In the *extensional* model, the event $R_{i,j} \in T$ is represented as the event $R_{i,j} \in T$ and for any k all alternative events $R_{i,k} \in T$ and $R_{j,k} \in T$ and all the events $R_{k,j} \in T$ and $R_{k,i} \in T$. The *mixed* is a combination of the other two models.

Experimental Evaluation

Here we want to demonstrate, with two sets of experiments, that our *induced* models can effectively exploit transitivity. The first experiment is a pilot experiment, the second experiment is a full experiment that differs from the pilot in the size of semantic networks and in target relations. For both sets of experiments we describe the experimental set up and we report the results.

The pilot experiment

In the *pilot experiment* we replicate a small existing semantic network of works with few pair of words in isa relation. To completely define the experiments we need to address some issues: how we defined the semantic networks to replicate, which corpus we have used to extract evidences for pairs of words, and which feature space and logit regressors we used. As corpus we used the *English Web as Corpus* (ukWaC) (Baroni, Berardini, Ferraresi, & and Zanchetta, 2009).

The best way of determining how a semantic network of words learner is performing is to see if it can replicate an existing semantic network. As target semantic networks we selected a portion of WordNet[iii] (Miller, 1995). Namely, we started from 44 concrete nouns divided in 3 classes: animal, artifact, and vegetable. For each word $\boldsymbol{w}$, we selected the synset $\boldsymbol{s_w}$ that generalizes the class it belongs to. In this way we obtained a set $\boldsymbol{S}$ of synsets. We then expanded the set to $\boldsymbol{S'}$ adding the siblings (i.e., the coordinate terms) for each synset in $\boldsymbol{S}$. The sets $\boldsymbol{S'}$ contains 265 coordinate terms plus the 44 original concrete nouns. For each element in $\boldsymbol{S}$ we collected the hypernyms, obtaining the set $\boldsymbol{H}$ of the hypernyms. We then removed from the set $\boldsymbol{H}$ the top classes (*entity*, *unit*, *object*, and *whole*), obtaining 77 hypernyms. For the purpose of the experiments we derived a taxonomy $\boldsymbol{T}$ from $\boldsymbol{S}$ and $\boldsymbol{S'}$ and a taxonomy $\overline{\boldsymbol{T}}$ from the set of negative examples. The taxonomy $\boldsymbol{T}$ is the portion of WordNet implied by $\boldsymbol{O} = \boldsymbol{H} \cup \boldsymbol{S'}$, i.e., $\boldsymbol{T}$ contains all the $(\boldsymbol{s},\boldsymbol{h}) \in \boldsymbol{O} \times \boldsymbol{O}$ that are in WordNet.

On the contrary, $\overline{\boldsymbol{T}}$ contains all the $(\boldsymbol{s},\boldsymbol{h}) \in \boldsymbol{O} \times \boldsymbol{O}$ that are not in WordNet. We then have 4596 positive pairs in $\boldsymbol{T}$ and 48354 negative pairs in $\overline{\boldsymbol{T}}$. To obtain the training and testing sets, we randomly divided the set $\boldsymbol{T} \cup \overline{\boldsymbol{T}}$in two parts, $\boldsymbol{T_{tr}} \cup \overline{\boldsymbol{T}}_{\boldsymbol{tr}}$ and $\boldsymbol{T_{ts}} \cup \overline{\boldsymbol{T}}_{\boldsymbol{ts}}$, respectively the 70% and 30% of the original $\boldsymbol{T} \cup \overline{\boldsymbol{T}}$.

We used a bag-of-n-gram feature space for implicitly modeling lexical-syntactic patterns. In learning process, we used a logistic regressor based on the Monroe-Penrose pseudo-inverse matrix (Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009).

Results

With the first set of experiments, we analyze the effectiveness of our *inductive* model with respect to the state-of-the-art. We evaluate the *iterative* probabilistic models (Snow, Jurafsky, & Ng, 2006), the *direct* probabilistic models (Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009), and the *induced* probabilistic models on their ability of sorting the pairs. We have two classes of methods. The *iterative* model adds some pairs at each step. The *direct* and the *inductive* probabilistic models, instead, produce a sorting of the pairs according to the probabilities.

We compared the two methods in the following way. For the *iterative* methods, we plot the curve that relates the accuracy to the number of added pairs. The accuracy is computed as the number of correctly added pairs with respect to the added pairs. On the contrary, for the probabilistic models we plot the accuracies with respect to the ranked pairs. For this set of experiments, we used k=100 for the pseudo-inverse matrix computation with SVD.

Figure 3: Accuracy of the top-k ranked pairs for the iterative, direct , and inductive probabilistic learners

The results are reported in Figure 3. Firstly, we can observe that, after some initial steps, models that keep the probabilities are better than the model that makes a decision at each step. The *direct* model already outperforms the *iterative* model. The second observation is that the *inductive* (*extensional*, *intensional*, and *mixed*) models outperform the *direct* model. This shows that our way of encoding the transitivity is effective. Finally, among the *inductive* models, the *mixed* model exploits both the *intensional* and *extensional* probabilistic definitions of concepts, proves to be the best one.

	Top k-pairs	
Probabilistic Model	*100*	*1000*
iterative	0.350	0.225
direct	0.290	0.269
intentional	0.510	0.282
extensional	0.420	0.292
mixed	0.510	0.322

Table 1: Accuracy of the different models at top 100 and 1000 ranked pairs

The accuracies are reported in Table 1. The table reports the accuracies for the different probabilistic models for two different cuts of the sorted pair list. The second and the third columns report, respectively, the accuracies for 100 and for 1000 considered pairs. We used these two cuts to compute the statistical significance of the difference between the direct and the mixed model. To determine the statistical significance, we used the model described in *(Yeh, 2000)* as implemented in *(Padó, 2006)*. We extended this latter for considering accuracies computed on sorted lists. According to these tests, the statistical significance is below 0.05.

The full experiment

Here, we want to demonstrate that our *induced* models can effectively exploit transitivity when increasing the size of the semantic network of both training and testing. Differently from the pilot experiment two target relations are considered: isa and part-of relations. To carry out the experiments we then need: (1) a corpus for extracting evidences to derive probabilities; (2) a semantic network of words and a set of negative examples for the target relation; (3) the definition of the feature space; and, finally, (4) the definition of the logistic regressors.

As corpus we used the *English Web as Corpus* (ukWaC) (Baroni, Berardini, Ferraresi, & and Zanchetta, 2009).

The semantic network of words will be used as source of training and testing examples. For each experiment we need: a training example set with positive pairs and negative pairs and a testing example set with positive pairs and negative pairs. The testing set TS should be a totally connected set for building the potential network of words. We want to test our model for two different transitive semantic relations: hyperonymy (*H*) and meronymy (*M*).

We extract the semantic networks and the set of negative examples from an existing knowledge repository, i.e., WordNet[iii] (Miller, 1995). In WordNet, semantic relations R are expressed as pairs of synonymy sets (synset), i.e., $R = \{(S_1, S_2) \mid S_1\ is\ in\ relation\ R\ with\ S_2\ \}$ where the synset S_1 and S_2 are the sets of words $S_1 = \{w_1^{(1)}, \dots, w_n^{(1)}\}$ and $S_2 = \{w_1^{(2)}, \dots, w_n^{(2)}\}$. The synset S_1 is in relation R with the synset S_2 if S_1 is directly related with the synset S_2 or if it is reachable with the transitive property. We derive the semantic networks of words from the synset network.

Given one of the two target relations, we can derive the network of words R from the set R as follows: $R = \{(w_a, w_b) | (S_a, S_b) \in R, w_a \in S_a, w_b \in S_b)\}$. We then derived the semantic networks of words for hyperonymy H and for meronymy M. These networks consist of, respectively, 7879350 and 672571 as reported in Table 2.

The negative examples have been obtained as follows. Given the set of the words in WordNet W, the negative examples are respectively $\bar{H} = W \times W - H$ and $\bar{M} = W \times W - M$.

Test	*Set*	*Description*	*Initial Size*	*Retrieved Pairs*
isa	TR_p	H/H_{ts}	1983197	212076

	TR_n	$\bar{H}/\bar{H}_{ts}$	5594387	315428
	TS_p	H_{ts}	506	150
	TS_n	$\bar{H}_{ts}$	80436	258
part-of	TR_p	M/M_{ts}	14333	8077
	TR_n	$\bar{H}/M_{ts}$	623616	318679
	TS_p	M_{ts}	408	101
	TS_n	$\bar{M}_{ts}$	34214	1713

Table 2: Semantic networks used in the experiments

For generating the testing set, we selected a relevant and strictly connected sub portion of network of words. This portion has been obtained using a synset as head and deriving the part of the network that can be transitively reached. For the H relation, we selected the sense 1 of “*vegetable*”. For the M relation, we selected the sense 1 of “*face*”. Given the sets $W(veg)$ and $W(face)$ of the words respectively in H_{ts} and M_{ts}, the negative examples are $\bar{H}_{ts} = W(veg) \times W(veg) - H_{ts}$, and $\bar{M}_{ts} = W(face) \times W(face) - M_{ts}$. In this way, we have the overall potential network of words for the testing.

The final sets are reported in Table 2. We here describe the two tests we made: the isa with *vegetable* and the part-of with *face*. The table reports how we obtained the positive examples and the negative examples for the training and the testing of the two examples. We also report the size of these sets and the number of the pairs retrieved in the corpus under the conditions later on described.

We used a bag-of-n-gram feature space for implicitly modeling lexical-syntactic patterns.

We used two different logistic regressors: a logistic regressor based on the Monroe-Penrose pseudo-inverse matrix (Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009) and the support vector machines (Vapnik, 1995) as implemented in (Joachims, 1999).

Results

In the first set of experiments, we want to investigate how *induced* model behaves with respect to the *direct* model in the most common settings for semantic relation learning: enriching an existing semantic network without any additional information. We then have the existing network out of which we can derive positive examples but also some negative example. We obtained this setting, that we call *semi-supervised*, using the two proposed sets for the two transitive relations. We gave an initial probability of 0.99 to the positive examples and of 0.5 for the negative examples. These latter are then used as if no information is available. This is the natural setting in learning semantic networks that is used in many experiments (e.g., (Pantel & Pennacchiotti, 2006)). The results of these experiments for the isa relation and the part-of relation are reported respectively in Table 3 and in Table 4. These tables report the *relative recall* of the different methods obtained using the first k ranked pairs. In line with (Pantel & Pennacchiotti, 2006), the *relative recall* RR is the ratio between the retrieved pairs with respect to the pairs that can be retrieved from the method, i.e., in our case the pairs that are retrieved in the corpus. In these tables, we report both the experiments with the pseudo-inverse matrix method (*PI*) and with SVM.

	direct		*intensional*		*extensional*		*mixed*	
	PI	SVM	PI	SVM	PI	SVM	PI	SVM
100	30.67	30.00	4.00	4.00	37.33	35.33	24.00	24.00
200	56.67	49.33	27.33	27.33	60.67	61.33	45.33	43.33
300	74.67	74.67	64.00	64.00	81.33	78.67	64.67	66.00

Table 3: Relative Recall of is-a relation: case semi-supervised

For each method, *direct*, *intentional*, and *extensional* we have the two columns representing the two methods for inducing the direct probabilities. For the isa relation (Table 3), we report the relative recall for the first 100, 200, and 300 first ranked pairs. For the part of relation(Table 4), we report the relative recall for 500 and 1000 first ranked pairs.

	direct		*Intensional*		*extensional*		*mixed*	
	PI	SVM	PI	SVM	PI	SVM	PI	SVM
500	28.71	28.71	32.67	32.67	33.66	33.66	34.66	33.64
1000	44.55	70.30	54.46	70.3	49.5	72.28	51.50	70.71

Table 4: Relative Recall of part-of relation: case semi-supervised

For the isa relation (Table 3Table 3), experiments show that the best way to exploit the transitivity of the isa relation is the *extensional* model. Only the *extensional* model outperforms the *direct* model. This is confirmed for both regression methods. We can also observe that the difference between the SVM and PI does not seem to be significant. For the part-of relation (Table 4)), experiments confirm that the *extensional* model outperforms the *direct* model. Yet, the *intensional* model behaves better than in the case of the isa relation.

To better explore our models, we then analyzed their behavior under ideal conditions. In this setting, we have explicit negative cases. Yet, these conditions hardly represent the operational scenario where the models act. Generally, we have an existing semantic network that we want to expand and we have no knowledge about negative examples. We obtained this setting, that we call *supervised*, assigning an initial probability of 0.99 to positive examples and an initial probability of 0.01 to negative examples. The results of these experiments for the isa and the part-of relations are reported respectively in Table 5 and in Table 6.

	direct PI	*intensional* PI	*extensional* PI	*mixed* PI
100	28.00	2.67	37.33	21.333
200	56.67	27.33	60.67	45.333
300	80.67	66.00	82.67	64.667

Table 5: Relative Recall of is-a relation Vegetable: case supervised

	direct PI	*intensional* PI	*extensional* PI	*mixed* PI
500	26.73	28.71	28.71	28.70
1000	39.60	49.50	44.55	46.51

Table 6: Relative Recall of part-of relation Face: case supervised

We report here the experiments for the pseudo-inverse method (PI). In the case of the isa relation, we can observe that this setting increases the performance only when we consider 300 pairs with respect to the semi-supervised approach. The *extensional* model is still better than the *intensional* model. For the part-of, the increase in performance with respect to the semi-supervised approach is lower than the previous case. Some part-of pairs that have been considered negative examples are positive. Inheritance of the part-of is not considered in generating positive examples. Yet, even in this case, the *extensional* model outperforms the *intensional* model. For the part-of relation, both the *intensional* and the *extensional* models are suitable for exploiting transitivity.

GENERIC ONTOLOGY LEARNERS ON APPLICATION DOMAINS

Domain knowledge bases are extremely important in a variety of natural language processing applications but manually creating structured knowledge repositories is a very time consuming and expensive task. Semi-supervised learning of domain knowledge bases from texts is generally seen as the solution. This is a very attractive and rich research area that is full of challenges. Generally, the process for automatically creating, adapting, or extending existing knowledge bases relies on existing structured knowledge and domain corpora. In ontology learning models using lexico-syntactic patterns (LSP) (Robison, 1970; Hearst, 1992; Pantel & Pennacchiotti, 2006), existing domain ontologies or structured knowledge bases give positive learning examples. These latter are exploited to learn lexico-syntactic patterns from domain corpora. Learnt LSPs are then used to extract and structure new knowledge from the domain corpora. For a successful application, these LSP methods for learning domain ontologies need large domain corpora and existing domain knowledge bases. LSP methods for learning ontologies from texts are good models only when we consider *ontology-rich* domains or we do generic knowledge extraction. In this latter case, these methods can exploit large general corpora and large general structured knowledge repositories such as WordNet (Miller, 1995). There are only few domains with well-assessed existing structured knowledge bases where the problem is to expand these ontologies. On the contrary, the large number of applications domains has little or no existing structured knowledge. The big challenge is to successfully apply these methods in *ontology-poor* domains. One of the possible ways to address the above challenge is to build LSP models that learn lexico-syntactic patterns on generic and ontology rich domains and then apply these patterns on specific ontology poor domains. In line with (Gao, 2009), we respectively refer as the *background domains* and *application domains* to these two kinds of domains. Yet, in machine learning and in statistical learning data should be enough representative of the environment where learned models will be applied. The statistical distribution of learning data should be similar to the distribution of the data where the learn model is applied. In this application scenario, this assumption is inaccurate. *Background domain data*, also called out-of-domain data, used for learning lexico-syntactic patterns have generally a different distribution with respect to *application domain data*, also called in-domain data. Generally, out-of-domain data are more than in-domain data. We need to envisage methods that exploit these data for building accurate in-domain models.

The rest of the section we present our model and then, we evaluate and assess the performance of our method on the target domain, i.e., Earth Observation Domain.

Learner Model: from Background to Application domain

Can training data from one corpus be applied to learn another corpus? The basic idea is partly to answer this question because we want to define an ontology learning model that can be adapted to previously unseen distributions of data. This model is thought to exploit the information learned in a *background* domain for extracting information in an *adaptation* domain.

Our ontology learning method is based on the probabilistic formulation given in the previous section. We use this probabilistic setting to learn a model that takes into consideration corpus-extracted evidences over a list of training pairs. The initial feature space is built starting from the analysis of a generic corpus where we observe a list of training pairs of words that are in a target semantic relation. We can generate these pairs using general resources such as WordNet. These pairs are used to enable the probabilistic method to induce lexico-syntactic patterns for the model of the specific semantic relation (Hearst, 1992). The learned model can be used to estimate the probabilities of the new instances computing a new feature space using the corpus of the *adaptation* domain.

In the rest of this section, we will firstly describe the background ontology learning model and we will then illustrate the method that we will be adapted to the new domain.

Background Ontology Learner

In the probabilistic formulation, the task of learning ontologies from a corpus is seen as a maximum likelihood problem. The ontology is seen as a set O of assertions R over pairs$R_{i,j}$. In particular we will consider the $is - a$ relation. In this case, if $R_{i,j}$ is in O, i is a concept and j is one of its generalizations. For example, $R_{dog,animal} \epsilon O$ states that *dog* is an *animal* according to the ontology O.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \epsilon O)$ of an assertion $R_{i,j}$ to belong to the ontology O and (2) the posterior probability $P(R_{i,j} \epsilon O | \vec{e}_{i,j})$ of an assertion $R_{i,j}$ to belong to the ontology O given a set of evidences $\vec{e}_{i,j}$ derived from the corpus. These evidences are derived from the contexts where the pair (i, j) is found in the corpus. The vector $\vec{e}_{i,j}$ is a feature vector associated to a pair (i, j). For example, a feature may describe how many times i and j are seen in patterns like "i *as* j" or "i *is a* j". But many other indicators exist of an Is-a relation between i and j (see (Hearst, 1992)). Given a set of evidences E over all the relevant word pairs, the probabilistic ontology learning task is defined as the problem of finding an ontology $\hat{O}$ that maximizes the probability of having the evidences of E, i.e.:

$$\hat{O} = \arg \overset{max}{O} P(E|O)$$

In the original model (Snow, Jurafsky, & Ng, 2006; Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009), this maximization problem was solved by a local search.

In the present model at each step we maximize the ratio between the likelihood $P(E|O')$ and the likelihood $P(E|O)$ where $O' = O \cup N$ and N are the relations added at each step. As in (Snow, Jurafsky, & Ng, 2006; Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009) this ratio is called $odds$. It is calculated using the logistic regression and then solving a linear problem using the pseudo-inverse matrix model. The regression coefficients will be estimated as follows

$$\hat{\beta} = X_{C_B}^{+} l$$

where l is the logit vector and $X_{C_B}^{+}$ is the **Moore-Penrose pseudoinverse** (Penrose, 1955) matrix of the inverse evidence matrix X_{C_B} obtained from a generic corpus C_B that includes a constant column of 1's, necessary to obtain the β_0 coefficients. The regressors represent the model that we learned from the training pairs using a generic corpus C_B that we will use to compute the probabilities of the testing pairs.

Estimator for Application Domain

In our task, instead of finding the ontology that maximizes the likelihood of having the evidences E, we calculate, given the regressors, the probabilities of the testing pairs step by step. The idea is that, given the domain based corpus C_A, for each testing pair we compute the vector space according to the features selected in the previous generic corpus feature space analysis. After the domain based corpus feature space analysis where we look for the testing pairs in C_A, we obtain a new feature space X_{C_A}. It is a matrix $n' \times m$ where n' is the number of the new instances found in the corpus C_A and m is the number of the features. We compute the logit of the new instances:

$$l' = \alpha X_{C_A} \hat{\beta} \qquad (2)$$

Where X_{C_A} is the inverse evidence matrix obtained from an *adaptation* domain corpus C_A that includes a constant column of 1's, necessary to obtain the β_0 coefficients. The parameter α is used to adapt the model by the β vector to the new domain. From the definition of logit we can compute the probabilities of the new instances, i.e.:

$$p_i = \frac{\exp(l_i)}{1 + \exp(l_i)}$$

This latter can be used to build the know ledge base in the new domain.

Experimental Evaluation

We experimented with our model adaptation strategy using a generic domain as *background* domain and the Earth Observation Domain as specific domain. We took the isa relation as the target relation. The target of the experiments is to understand whether or not our model adapt to specific domains. We then compare our system (Our-System) with respect to a system that uses only WordNet (WN-System). In this section, we firstly describe the general experimental set up. We then describe the quality of the target domain ontologies. Finally, we analyze the accuracy of our models with respect to the three different ontologies.

Experimental Setup

To define completely the experiments we have to define: both training and testing pairs, which corpus has been used to extract evidences for training pairs, which corpus to extract evidences for testing pairs, and which feature space we use for both corpora. To build the training pairs we generated all the pairs that were in hyperonym relation in WordNet[iii] (Miller, 1995) and we obtained about 2 millions of words.

Here, we firstly define the semantic networks used in the experiments. The network of words will be used as a source of training and testing examples. For each experiment we need: a training example set $TR = (TR_p, TR_n)$ with positive pairs and negative pairsTR_p, and a testing example set TS. To build TS we start from a given list of 63 terms that are relevant in Earth Observation Domain. Then we combine each term with the other terms and we generate 63×63 pairs. Furthermore, for each term w, we select all the synsets s_w in WordNet. In the case of a term with a synset in WordNet we generate the pairs combining w with all the hyperonyms for each synset. Otherwise, if w has compound words we look for our semantic head in WordNet. If we find the synsets, we generate the pairs combining w with the hyperonyms of the semantic head of w.

We extract the training example pairs from an existing knowledge repository: WordNet[iii] (Miller, 1995). Given hyperonymy as target relation, we can derive the network of words R from the set R as follows:$R = \{(w_a, w_b) | (S_a, S_b) \in R,\ w_a \in S_a, w_b \in S_b\}$. We then build the set H that contains all pairs of words in WordNet that are in hyperonymy relation. Then $TR_p = H - TS$. Given the set of the words in WordNet W, the training negative example is $TR_n = W \times W - TR_p - TS$. We build TR_p , TR_n and TS without overlap. We searched for the pairs in TR in a corpus C_B (in particular the *English Web as Corpus* (ukWaC) (Baroni, Berardini, Ferraresi, & and Zanchetta, 2009) has been used). This is a web extracted corpus of about 2700000 web pages containing more than 2 billion words. It contains documents of several different topics such as web, computers, education, public sphere, etc.. It has been largely demonstrated that the web documents are good models for natural language (Lapata & Keller, 2004).

Using a web crawler, here we pick up a corpus related to Earth Observation Domain C_A, successively ” cleaned” , that contains about 8300 documents (115,6 MB). We use the bag-of-word feature space. Out of the $T \cup \bar{T}$, only those pairs that appeared at a distance of 3 tokens at most have been selected. Using these 3 tokens, we generate the *bag-of-word* feature space. The pairs in TR found in the ukWaC are 527348, while the pairs in TS found in are 404. The two generated feature spaces have the same features that are 276670. The model to build ontologies in Earth Observation Domain has been generated by using the training pairs and the corpus ukWac.

Evaluating the Quality of Target Domain Specific Ontologies

We want to evaluate our approach in learning the bulk of the ontologies, i.e., the *isa* relation, in Earth Observation Domain. between two pairs of words is a binary problem. We then asked three annotators (A_1, A_2 and A_3) to build three different ontologies: two of them are expert in the domain ($A_1$and A_2), the third one is not (A_3). A_1 and A_2 have different levels of expertise: A_1 is a young expert in the domain and A_2 an older one. Each annotator made a binary classification of 641 pairs of words in Earth Observation Domain, i.e., the TS set introduced in the previous section.

We then wanted to judge the quality of the annotation procedure according to their inter annotation agreement. A simple measure of the quality of the agreement rate between two human annotators is the ratio between the number of items identically judged by two different annotators and the total number of items considered by the annotators. In (Scott, 1955), this measure is named **observed agreement** and it is defined as *the percentage of judgments on which the two analysts agree when coding the same data independently*. In accord to (Artstein & Poesio, 2008) we define the agreement value.

We can examine the issue of inter-annotator agreement by comparing the agreement rate of the human annotators. There are different methods for measuring the agreement among 3 annotators. When there are more than two annotators, some of them may agree and the rest disagrees on the same item. In this case, the observed agreement can no longer be defined as the percentage of items getting agreement. To solve this problem , we can analyze two solutions : $\boldsymbol{pairwise\ agreement}$ and $multi - \boldsymbol{\pi\ agreement}$ both in (Fleiss et al., 1971). In the section Pairwise agreement we will describe the inter-annotators agreement for each pair of annotators that has a personal distribution and we will show that this is similar to the distribution computed on both annotators of each pair. In the $multi - \boldsymbol{\pi}$ agreement, we examine the distribution of all the three annotators.

Pairwise agreement

The pairwise agreement defines the agreement on a particular item as the proportion of agreed judgment pairs out of the total number of judgment pairs for that item (Fleiss, et al., 1971). We measure the inter-annotators agreement of the 3 pairs of annotators: $pair_1$ for the two annotators expert in the domain A_1 and A_2 ; $pair_2$ for one annotator expert in the domain A_1 and the other one not expert A_3; and, $pair_3$ for the second annotator expert in the domain A_2 and the other one not expert A_3.

		A_1		
		yes	no	
A_2	yes	47	61	108
	no	43	490	533
		90	551	641

(a) $pair_1$=(A_1,A_2)

		A_1		
		yes	no	
A_3	yes	76	83	159
	no	14	468	482
		90	551	641

(b) $pair_2$=(A_1,A_3)

		A_2		
		yes	no	
A_2	yes	72	87	159
	no	36	446	482
		180	553	641

(c) $pair_3$=(A_2,A_3)

Table 7:Contingency tables for pairwise annotator agreement for 641-annotations

	Ao	Ae	$Kappa$
$pair1 = (A1, A2)$	0.8377535	0.7384206	0.3797428
$pair2 = (A1, A3)$	0.8486739	0.6811997	0.5253266
$pair3 = (A2, A3)$	0.8081123	0.6670496	0.4236749

Table 8: pairwise agreement for 641-annotationsions

Given the same data (641 or 404-annotations) with the same guidelines, we build the contingency tables for the 3 pairwise annotators (respectively Table 7 and Table 9). For each table we report the statistic of the two annotators. Then in Table 7 (a) we summarize the inter-annotator agreement of the 3 pairwise agreements considering 641-annotators. For example, the observed agreement for this data is obtained summing up the cells of the table where the annotators assign the same judgment and dividing by the total number of annotations. For example, considering $pair_1$ (first row of the Table 8(a)), the two annotators label 47 occurrences as YES, and 490 as NO. The resulting observed agreement of $pair_1$ is $A_o = (47 + 490)/641 = 0.8377535$. As above mentioned, there are two different methods to compute the expected agreement. In the first method the expected agreement is governed by prior distributions that are unique for each annotator and it is computed looking the actual distribution. Then for $pair_1$ we have$A_e = 0.16848674 * 0.1404056 + 0.83151326 * 0.8595944 = 0.7384206$.

In the second method we get the same distribution for each annotator of the *pair*, then we have

$$A_e = \left(\frac{90 + 108}{641 * 2}\right)^2 + \left(\frac{533 + 551}{641 * 2}\right)^2 = 0.7388149$$

Since the two A_e values are similar and the same occurs for the other pairs, we report only the expected agreement computed using the first method.

Finally, using both the observed and expected agreement, the possible agreement beyond change observed for the is $kappa = (0.83775350.7384206)/(1 - 0.7384206) = 0.3797428$. Analogously we compute kappa value for the other pair of annotators.

In the same way we compute Observed Agreement, Expected Agreement and coefficient kappa for the pairwise agreement considering 404-annotations (Table 10). Summarizing only for on 641-annotations the coefficient kappa is in the “fair” interval in accord to the scale proposed in (Landis & Koch, 1977). Most likely there is a fair agreement between annotators A_2 and A_3 because the first one is an older expert in the domain while the second one is not expert at all, so they have a different knowledge with respect to the specific Earth Observation Domain.

In all the other cases the pairwise agreement is better because the coefficient kappa belongs to the “moderate” interval. We are confident on the reliability of such annotations as the annotators agree on labeling the same pairs of words. This allows us to prove the validity of the annotation.

		A_1		
		yes	no	
A_2	yes	40	61	108
	no	43	490	533
		90	551	641

(a) pair$_1$=(A$_1$,A$_2$)

		A_1		
		yes	no	
A_3	yes	76	83	159
	no	14	468	482
		90	551	641

(b) pair$_2$=(A$_1$,A$_3$)

		A_2		
		yes	no	
A_2	yes	72	87	159
	no	36	446	482
		180	553	641

(c) pair$_3$=(A$_2$,A$_3$)

Table 9: Contingency tables: pairwise annotator agreement for 404-annotations

	Ao	Ae	$kappa$
$pair1 = (A_1, A_2)$	0.8341584	0.7023086	0.4429077
$pair2 = (A_1, A_2)$	0.8415842	0.6291663	0.5728117
$pair3 = (A_2, A_3)$	0.7896040	0.6322174	0.4279336

Table 10: pairwise agreement for 404-annotations

Multi-π agreement

In $multi - \pi$ agreement the agreement of the annotators is considered as a whole. There is only one distribution for all the annotators, derived from the total proportions of categories assigned by each annotator.

When there are more than two annotators, the visualization of the data is a difficult task: a possible solution is in using the agreement table where each annotator is represented in a separate column.

The columns A_1, A_2 and A_3 of Table 11(a) Table 11(b) the label 1 or 0 assigned for each pair (first column) by the 3 annotators respectively in 641 or 404-annotations. For both tables we report in the columns YES and NO respectively the sum of 1s and 0s in A_1, A_2 and A_3. In Table 11(c) we report the observed and expected agreement and the relative kappa coefficient for both 641 and 404 annotations. The kappa value obtained from both annotations confirms the conclusions deduced with the pairwise agreement method that proved the validity of the annotations of the 3 annotators.

pairs of words	A_1	A_2	A_3	**Yes**	**NO**
(agriculture,department)	0	0	0	0	3
(soil,earth)	1	1	1	3	0
(agriculture,business)	0	0	0	0	3
(wind,direction)	1	0	0	1	2
(climate,climate change)	0	0	0	0	3
(climate change,climate)	0	1	1	2	1
(climate change,activity)	1	0	1	2	1
(forest,terra firma)	1	1	1	3	0
…	…	…	…	…	…
TOTAL	90	108	159	357(0.19)	1566(0.81)

(a) Agreement table for 641-annotations

pairs of words	A_1	A_2	A_3	**Yes**	**NO**
(forest,terra firma)	1	1	1	3	0
(wind,process)	0	0	0	0	3
(forest,object)	0	0	0	0	3
(cloud,state)	0	1	0	1	2
(soil,object)	0	1	1	2	1
(wind,breath)	0	0	0	0	3
(wind,act)	0	0	0	0	3
(topography,geography)	1	1	1	3	0
…	…	…	…	…	…
TOTAL	75	72	119	266(0.22)	946(0.78)

(b) Agreement table for 404-annotations

	Ao	Ae	$kappa$
641-annotations	0.83151	0.69764	0.44277
404-annotations	0.82382	0.65739	0.48577

(c) Multi-π agreement respect to 641 and 404 annotations

Table 11:Agreement tableand Multi-π agreement for 641 and 404 annotations

Result

In our experiments we investigated how the approach to compute a model using both a *background* domain and an existing network, can be positively used to learn the *isa* relation in Earth Observation Domain. For the evaluation, we compare our learner model (*Our-System*) directly with currently existing hyperonym links in WordNet (*WN-System*) and we measure in both cases the performance to find correctly the testing pairs that are in isa relation. In order to evaluate the performance of the two systems for the pairs in Earth Observation Domain we used the three different ontologies produced by the three annotators. We will call these three target ontologies with the name of the annotator.

The results of the experiments are reported in Table 12(a) and in Table 12(b). In the first table we compute the recall, the precision and the f-measure of the *WN-System* against the 3 ontologies, while in the second table we compute the recall, the precision and the f-measure of the *Our-System.*

annotators	recall	precision	f-measure
A_1	0,36	0.184932	0,244344
A_2	0,305556	0,150685	0,201836
A_3	0,470588	0,383562	0,422642

(a) WN-System against the 3 annotators

annotators	recall	precision	f-measure
A_1	0,493333	0,253425	0,334842
A_2	0,4305556	0,212329	0,284404
A_3	0,4369748	0,356164	0,392453

(b) Our-System against the 3 annotators

Table 12:Performance of both system with respect to 3 annotators

We can then draw some observations: First, *Our-System* behaves better than the *WN-System* on the ontologies produced by the expert annotators. The f-measure of both the expert annotators (A_1 and A_2) is better for *Our-System* with respect to *WN-System*. On the contrary, for the last ontology (A_3) the *WN-System* has better performance than our system. Then, our system is capturing knowledge of the specific domain as it is behaving better than the generic system with respect to domain experts. Second, in the case of the expert annotators, the recall of our system is higher than the recall of the WordNet based system. This confirms that the coverage of WordNet in the specific domain is low and only learning methods can be used to adapt the ontological information to the specific domain. On the contrary, for the non-domain expert, WordNet is good enough to cover domain knowledge. Results show that *Our-System* is a good learner method that can be positively used to learn the *isa* relation in Earth Observation Domain.

PROBABILISTIC ONTOLOGY LEARNER IN SEMANTIC TURKEY

Ontologies and knowledge repositories are important components in Knowledge Representation (KR) and Natural Language Processing (NLP) applications. Yet, to be effectively used, ontologies and knowledge repositories have to be large or, at least, adapted to specific domains. Even huge knowledge repositories such as WordNet (Miller, 1995) are extremely poor when used in specific domains such as the medical domain (see (Toumouth, Lehireche, Widdows, & Malki, 2006)).

In automatically creating, adapting, or extending existing knowledge repositories using domain texts is a very important and active area a large variety of methods have been proposed: ontology learning methods in KR (Medche, 2002; Cimiano, Hotho, & Staab, Learning concept hierarchies from text corpora using formal concept analysis, 2005; Navigli & Velardi, 2004) as well as knowledge harvesting methods in NLP either (Hearst, 1992; Pantel & Pennacchiotti, 2006).These learning methods use variants of the distributional hypothesis or exploit some induced lexical-syntactic patterns (Robison, 1970). The task is generally seen as a classification (e.g., (Pekar & Staab, 2002; Snow, Jurafsky, & Ng, 2006)) or a clustering (e.g., (Cimiano, Hotho, & Staab, Learning concept hierarchies from text corpora using formal concept analysis, 2005)) problem. This allows the use of both machine learning and probabilistic models. But generally, automatic models for extracting ontological knowledge from texts do not have the performance needed to extend existing ontologies with a high degree of accuracy. As a consequence, the resulting automatically expanded ontologies can be completely useless. Generally, systems for augmenting ontologies extracting information from texts foresee a manual validation for assessing the quality of ontology expansion. Yet, these systems do not use the manual validation for refining the information extraction model that proposes novel ontological information. Here, the idea is to prefer methods that can use decisions of final users to incrementally refine the model for extracting ontological information from texts, i.e., each decision of final users is exploited in refining the parameters of the extraction model. Including these new examples as training for machines helps in augmenting the performances of the automatic extractor, as shown in (Cimiano & Volker, Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery, 2005). In the following, we present the Semantic Turkey Ontology Learner (ST-OL) (Fallucchi, Scarpato, Stellato, & Zanzotto, 2009), an incremental ontology learning system that follows the above idea putting final users in the learning loop. Furthermore, this system uses the proposed probabilistic ontology learning model that exploits transitive relations for inducing better extraction models.

The chapter is organized as follows. We firstly present the ideas behind our new ontology learning system introducing the concept of incremental ontology learning. We then introduce ST-OL, the system that we have adopted following the above principles. Finally, we draw some conclusions.

Incremental Ontology Learning

To efficiently set-up an incremental model for ontology learning, we have to address two issues:

- we need an efficient way to interact with final users
- we need an incremental learning model

The rest of the section shows how we can address these issues using existing models and existing systems. We start from presenting the concept of incremental ontology learning. Then, we describe the used ontology editor and finally, we introduce the adopted ontology learning methodology.

The concept

The incremental ontology learning process we want to model leverages on the positive interaction between an automatic model for *ontology learning* and the final users. We obtain this positive interaction using one additional component: an *ontology editor*. The overall process is organized in two phases: (1) the *initialization step* and (2) the *learning loop*. In the *initialization step*, the user selects the initial ontology and the corpus. The system, then, uses these two elements to generate the first model for learning ontological information from documents. In the *learning loop*, the machine learning component extracts a ranked list of pairs ($candidate\_concept, superconcept$) and the user selects, among the first k pairs, the correct ones to be added to the ontology. We then use these choices to generate both positive and negative training examples for the ontology learning component. Once the new ontology extraction model has been learnt (using the corpus, the updated ontology, and the growing *non-ontology*), the process restarts from the beginning of the loop.

Given a selected corpus C, the initial ontology O_0 and the generic ontology O_i at the iteration i, we can see the incremental learning process as the sequence of the resulting ontologiesO_0 ... O_n.The *transition* function leverages on the ontology learning model M and on the interaction with the user, i.e., the user validation UV. This function can be represented as follows:

$$M_C(O_i, \bar{O}_i) = \hat{O}_{i+1} \overset{\text{UV}}{\rightsquigarrow} (O_{i+1}, \bar{O}_{i+1}) \qquad (3)$$

where M_C is the model learnt from the corpus, O_i, is the ontology at the $i-th$ step and $\bar{O}_i$ are the negative choices of the users at the same step. This model gives as output a ranked list of possible updates of the ontology $\hat{O}_{i+1}$. The UV on the first k possibilities produces the updated ontology and the updated *non-ontology* $\bar{O}_{i+1}$. At the initial step, the process has O_0 and $\bar{O}_0 = \emptyset$. The *ontology learner* produces the model $M_C(O_i, \bar{O}_i)$ building feature vectors representing the contexts of the corpus C where we can find pairs of pairs (*candidate_concept, superconcept*). These pairs are extracted from the ontology O_i,and the *non-ontology* $\bar{O}_i$.

Semantic Turkey

Semantic Turkey is a Knowledge Management and Acquisition system developed by the Artificial Intelligence Group of the University of Rome, Tor Vergata. Semantic Turkey (ST, from now on) was initially developed as a web browser extension (it is currently implemented for the popular Web Browser Mozilla Firefox) for *Semantic Bookmarking* (Griesi, Pazienza, & Stellato, Gobbleing over the Web with Semantic Turkey, 2006), that is, the process of *eliciting* information from (web) documents, to *acquire* new knowledge and *represent* it through representation standards, while *keeping reference* to its original information sources.

Semantic Bookmarks are different from their traditional cousins because they abandon the purely portative semantics of traditional links&folders bookmarking, and promote a new paradigm, aiming at "a clear separation between (acquired) knowledge data (the WHAT) and their associated information sources (the WHERE)". In practice, the user is able to select portions of text from web pages loaded from the browser, and to annotate them in an (user defined) ontology. A neat separation is maintained between the ontological resources created from the annotation, and the annotations themselves. In this way, the user can easily organize the knowledge (by establishing relationships between ontology objects, categorizing them, better defining them through attributes etc...), while keeping multiple bookmarks in a separated space, pointing to ontology resources and carrying with them all information related to the taken annotations (such as the page where the annotation has been taken, its title, the text which was referring to the created/referenced ontology resource etc...). Easy-to-perform drag-and-drop operations were thought to optimize user interaction, concentrating the creation of both the ontological resources and their related annotations in a few mouse clicks.

ST has lately evolved (Griesi, Pazienza, & Stellato, Semantic Turkey - a Semantic Bookmarking tool (System Description), 2007) in a complete Knowledge Management and Acquisition System based on Semantic Web technologies, introducing full support for ontology editing and improving functionalities for annotation&creation, ST has explored a new dimension without predecessors in the field of Ontology Development or Semantic Annotation, unique in the process of building new knowledge while exploring the web. The new objective of ST has been thus reducing the impedance mismatch between domain experts and knowledge investigators on one side, and knowledge engineers on the other side, providing a unifying platform for acquiring, building up, reorganizing and refining knowledge. The ontology learning module that we introduce here has been implemented and integrated upon the above exposed framework.

Probabilistic Ontology Learner

We use the proposed probabilistic ontology learning (POL) to expand existing ontologies with new facts. In POL it is possible to take into consideration both corpus-extracted evidences and the structure of the generated ontology. In the probabilistic formulation (Snow, Jurafsky, & Ng, 2006), the task of learning ontologies from a corpus is seen as a maximum likelihood problem. The ontology is seen as a set O of assertions R over pairs . In particular we will consider the *is-a* relation. In this case, if is in O, i is a concept and j is one of its generalizations (i.e., the direct or the indirect generalization). For example, describes that *dog* is an *animal* according to the ontology O.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \epsilon\ O)$ of an assertion $R_{i,j}$ to belong to the ontology O and (2) the posterior probability $P(R_{i,j} \epsilon\ O \mid \overrightarrow{e_{i,j}})$ of an assertion $R_{i,j}$ to belong to the ontology O given a set of evidences $\overrightarrow{e_{i,j}}$ derived from the corpus. These evidences are derived from the contexts where the pair (i,j) is found in the corpus. The vector $\overrightarrow{e_{i,j}}$ is a feature vector associated with a pair (i,j). For example, a feature may describe how many times i and j are seen in patterns like "$i\ as\ j$" or "$i\ is\ a\ j$". These, among many other features, are indicators of an $Is - a$ relation between i and i (see (Hearst, 1992)).

Given a set of evidences E over all the relevant word pairs, in (Snow, Jurafsky, & Ng, 2006) the probabilistic ontology learning task is defined as the problem of finding an ontology O that maximizes the probability of having the evidences E, i.e.:

$$\hat{O} = arg_{O}^{max}\ P(E|O) \qquad (4)$$

In the original model (Snow, Jurafsky, & Ng, 2006), this maximization problem is solved with a local search. In the incremental ontology learning model that we propose, this maximization function is solved using also the information coming from final users.

In the user-less model, what is maximized at each step is the ratio between the likelihood $P(E|O')$ and the likelihood $P(E|O)$ where $O = O \cup N$ and N are the relations added at each step. This ratio is called multiplicative change Δ(N) and is defined as follows:

$$\Delta(N) = \frac{P(E|O')}{P(E|O)}$$

It is also possible to demonstrate that

$$\Delta(R_{i,j}) = k \cdot \frac{P(R_{i,j} \in O|\overrightarrow{e_{i,j}})}{1 - P(R_{i,j} \in O|\overrightarrow{e_{i,j}})} = k \cdot odds(R_{i,j})$$

where k is a constant (see (Snow, Jurafsky, & Ng, 2006)) that will be neglected in the maximization process.

We calculate the $odds$ using the logistic regression. The regression coefficients can be estimated using the Monroe-Penrose pseudo-inverse matrix (Fallucchi & Zanzotto, SVD Feature Selection for Probabilistic Taxonomy Learning, 2009)

$$\hat{\beta} = X^{+} l \qquad (5)$$

where β is an approximation of the regression coefficients vector, X^{+} is the inverse evidence matrix, and l the logit vector.

In our user-oriented incremental ontology learning model we propose to include final users in the loop. In our task we do not find the ontology that maximizes the likelihood of having the evidences E. We calculate the probabilities step by step. Then we present an ordered set of choices to final users that make the final decision on what to use in the next iteration. The order set is obtained using the logit function as it is equivalent to the order given by the probabilities. For this reason, in the following we will operate directly on the logit rather than on the probabilities. It is possible to calculate the logit vector at the $i - th$ iteration using the logit definition and the equation β=$X^{+} l$ (5):

$$XX^{+}l_i = \hat{l}_{i+1} \overset{UV}{\leadsto} l_{i+1} \qquad (6)$$

At each iteration, we calculate the logit vector using the logit vector of the previous iteration. The logit vector is then changed in the user validation (UV). When the user accepts a new relation its probability is set to 0.99. On the contrary, when the user discards a relation its probability is set to 0.01. The matrix XX^{+} is constant for each iteration. In particular, we have found a matrix XX^{+} that is the constant model M_C of the equation *MC(Oi,Oi)=Oi+1* UV↝*Oi+1,Oi+1* (3). The matrix depends only on the corpus C and not on the initial ontology. The logit vector l represents both the current ontology O_i and the negative ontology $\bar{O}_i$ as it includes the logit of both probabilities (0.99 and 0.01).

Semantic Turkey-Ontology Learner (ST-OL)

The model described in previous section has been implemented and integrated in a Semantic Turkey extension called ST Ontology Learner (ST-OL). ST-OL provides a graphical user interface and a human-computer interaction work-flow supporting the incremental learning loop of our learning theory. If the user has loaded an ontology in ST, he can to improve it by adding new classes and new instances using ST-OL. The interaction process is achieved through the following steps:

- an *initialization phase* where the user selects the initial ontology O and the bunch of documents C where to extract new knowledge
- an *iterative phase* where the user launch the learning and validates the proposals of ST-OL

Thus, starting from the initial ontology O and a bunch of documents C, the user has the possibility of using an incremental ontology learning model.

For the *initialization phase*, the User Interface (UI) of ST-OL allows users to select the initial set of documents C (corpus), and to send both the ontology O and the corpus C to the learning module. To start this stage of the process, the user selects *"Initialize POL"* on the ST-OL panel (see Figure 4). The probabilistic ontology learner analyzes the corpus, finds the contexts for each ontological pair, computes the first extraction model, and, finally, proposes the pairs that are in is-a relation. This first analysis is the most expensive, because devoted to computing the matrix XX^{+}. Yet, this computation is done only once in the iterative process.

Once this initialization finishes, the *iterative phase* starts. ST-OL enables the button labeled *"Proposed Ontology"*. The effect of this button is to show the initial ontology extended with the pairs proposed by POL. Figure 4 shows an example of an enriched initial ontology.

Figure 4: Initial Ontology extended with the pairs proposed by the POL System

The main goal of ST-OL is draw the attention to the good added information. The user has the possibility of selecting the pairs he wants to add among the proposed pairs. To drive the attention towards the good pairs, we use different brightness of red for the different probabilities. More intense tonalities of red represent higher probabilities.

Figure 5: Manual validation of new resources added to the ontology

In order to focus, if possible, only on good pairs, ST-OL shows only pairs above a threshold τ of probabilities. For example, Figure 4 the relation (i.e., the pair) between "truck" and "container" is more probable than the relation between "spreader" and "container". Then different red tones are used. At this point, the user can accept or reject the information. After acceptance, the new information is stored in the ST ontological repository and can be browsed as usual through the ontology panel on the Firefox sidebar.

Figure 5 shows what happened when the user accepted two proposed pairs: “mango” as instance of “fruit” and “pepper” as subclass of “vegetable”.

In the incremental model the above activity enables to build an upgraded probability vector. When the user accepts a new pair, ST-OL updates its probability to 0.99. When the user discards the pair, its probability is set to 0.01. These new values are used for the next iteration of the leaning process. After some manual evaluation, the user can decide to update the proposed ontology. Given the probabilistic ontology learning model, this new evaluation is just a simple multiplication between the existing matrix XX^+ and the new vector. To force the recompilation, the user can use the “*Proposed Ontology*” button.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Describing word meaning is one of the most interesting challenges of natural language processing as texts cannot be "understood" without a clear and formal model of its basic components. Semantic networks of words are often used as formal models of word meaning but, to be useful for final NLP applications, these networks should large enough to cover words used in the final domain of the applications. It is nearly impossible to manually obtain a wide coverage for these semantic networks. Automatically learning these semantic networks from domain corpora is then the preferred solution. Models for automatically expanding semantic networks of words from texts use corpus-extracted evidences to determine whether or not new pairs of words are in a given semantic relation and, then, have to be included in existing knowledge repositories. These decision systems are trained observing how pairs of words in a given semantic relation behave in document collections. This information is used to induce a model that is then applied to novel word pairs. This chapter has explored this important area of research giving important contributions and advancing state-of-art models.

First, we observed that structural properties of semantic networks of words, when relevant, are not used in machine learning models to better induce relevant features to determine confidence values for extracting semantic relations. Semantic relation learning models based on the distributional hypothesis, for example, use the structural properties of semantic networks of words such as transitivity only intrinsically, but they cannot be applied for learning transitive semantic relations other than the generalizations. Even where transitivity is explicitly used, it is not directly exploited to model confidence values. On the contrary, LSP models can learn any kind of semantic relations but they do not explicitly exploit the structural properties of target relations when learning taxonomies or semantic networks of words. We have demonstrated that keeping the probability within the final knowledge base is extremely important for the performances of the learning method as it gives the possibility to better use structural properties of target relations such as transitivity. Our probabilistic model is suitable for exploiting the structural properties of semantic relations in learning semantic networks.

Second, we observed that systems that automatically create, adapt, or extend existing semantic networks of words need a sufficiently large number of documents and existing structured knowledge to achieve reasonable performance. If the target domain has not relevant pre-existing semantic networks of words, we will not have enough data for training the initial model. Obtaining manually structured knowledge repositories in specific domains is a very time consuming and expensive task. We have shown that our learning method that exploits the models learned from a generic domain is helpful to discover the relation between two words in a specific domain. Our learning model exploits training data for building in-domain models with bigger accuracy with a very small effort for the adaptation to different specific knowledge domains.

Finally, we studied models to include the manual validation for assessing the quality of semantic networks of words expansion within systems for creating or augmenting semantic networks of words . ST-OL provides a graphical user interface and a human-computer interaction work-flow supporting the incremental learning loop of our probabilistic learning models. This system efficiently interacts with final users exploiting an incremental model that in learning loop includes final users. The probabilistic model is integrated in a Knowledge Management and Acquisition platform Semantic Turkey. Thus, ST-OL has proven to be the right environment for embodying this kind of process, providing the crossroads between Users, Web and Knowledge.

In the future, a natural improvement is the analysis of different and more informative feature spaces such as those based on syntactic models. We believe this will boost the performances of our model. We have here shown that the model can be applied to different transitive relations (i.e., isa and part-of). Yet, we need to explore different transitive semantic relation, e.g., cause-effect, entailment and we plan to extend the model to consider other structural properties of semantic networks.

REFERENCES

Artstein, R., & Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Journal of Computational Linguistics* , 34 (4).

Bacchiani, M., Roark, B., & Saraclar, M. (2004). Language Model Adaptation with MAP Estimation and the Perceptron Algorithm. In D. M. Susan, & S. Roukos (Editors), *Proceedings of HLT-NAACL: Short Papers* (p. 21-24). Association for Computational Linguistics.

Bagni, D., Cappella, M., Pazienza, M. T., Pennacchiotti, M., & Stellato, A. (2007). Harvesting Relational and Structured Knowledge for Ontology Building in the WPro architecture. In R. Basili, & M. T. Pazienza (Editors), *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence on AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*. 4733, p. 157-169. Springer.

Baroni, M., Berardini, S., Ferraresi, A., & and Zanchetta, E. (2009). The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation* , 43, 209-226.

Basili, R., Gliozzo, A., & Pennacchiotti, M. (2007). Harvesting Ontologies from Open Domain Corpora: a Dynamic Approach. *Proceedings of Recent Advances on Natural Language Processing*. Borovets, Bulgaria.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American* , 284 (5), 34-43.

Bertoldi, N., & Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. *StatMT '09 Proceedings of the Fourth Workshop on Statistical Machine Translation* (p. 182-189). Association for Computational Linguistics.

Blitzer, J., Mcdonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. *EMNLP '06 Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. *Proceedings of the 1st European Semantic Web Symposium (ESWS)*.

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* , 22 (2), 249-254.

Chan, Y. S., & Ng, H. T. (2007). Domain Adaptation with Active Learning for Word Sense Disambiguation. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (p. 49-56). Association for Computational Linguistics.

Chelba, C., & Acero, A. (2006). Adaptation of Maximum Entropy Capitalizer: Little Data Can Help a Lot. *Computer Speech & Language* , Vol. 20 (No. 4), 382-399.

Chklovski, T., & Pantel, P. (2004). VerbOCEAN: Mining the Web for FIne-grained Semantic Verb Relations. *Proceedings of EMNLP 2004*. Association for Computational Linguistics.

Cimiano, P., & Volker, J. (2005). Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In A. Montoyo, R. Munoz, & E. Metais (Editors), *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*. 3513, p. 227-238. Springer.

Cimiano, P., Hotho, A., & Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence research* , 24, 305-339.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Psychological Bulletin* , 20, 37-46.

Fallucchi, F., & Zanzotto, F. M. (2010). Inductive Probabilistic Taxonomy Learning using Singular Value Decomposition. In *Journal of Natural Language Engineering* .

Fallucchi, F., & Zanzotto, F. M. (2009). SVD Feature Selection for Probabilistic Taxonomy Learning. *GEMS '09 Proceedings of the Workshop on Geometrical Models of Natural Language Semantics* (p. 66-73). Association for Computational Linguistics.

Fallucchi, F., Scarpato, N., Stellato, A., & Zanzotto, F. M. (2009). Probabilistic Ontology Learner in Semantic Turkey. *AI*IA '09: Proceedings of the XIth International Conference of the Italian Association for Artificial Intelligence Reggio Emilia on Emergent Perspectives in Artificial Intelligence* (p. 294-303). Springer-Verlag.

Fleiss, J., & others. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin* , 76 (5), 378-382.

Gao, J. (2009). Model adaptation via model interpolation and boosting for web search observations of markov chains. *IEEE Transations on Speech and Audio Processing* (Vol. 2, 291-298)..

Geffet, M., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. *Proceedings of The 43rd Annual Meeting of the Association for Computational Linguistics* (p. 107-114). Association for Computational Linguistics.

Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubzy, M., & Eriksson, H. (2003). The evolution of Protégé-2000: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies* , 58 (1), 89-123.

Gildea, D. (2001). Corpus Variation and Parser Performance. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing EMNLP* .

Gómez-Pérez, A., & Manzano-Macho, D. (2003). Deliverable 1.5: A survey of ontology learning methods and techniques.

Griesi, D., Pazienza, M. T., & Stellato, A. (2006). Gobbleing over the Web with Semantic Turkey. In *Proceedings of SWAP'2006*

Griesi, D., Pazienza, M. T., & Stellato, A. (2007). Semantic Turkey - a Semantic Bookmarking tool (System Description). In E. Franconi, M. Kifer, & W. May (Editors). *The Semantic Web: Research and Applications*, 4519, p. 779-788. Springer.

Haase, P., Lewen, H., Studer, R., Tran, D. T., d'Aquin, M., & Motta, E. (2008). The NeOn Ontology Engineering Toolkit.

Harris, Z. (1964). Distributional Structure. In J. J. Katz, & J. A. Fodor (Editors), *The Philosophy of Linguistics*. Oxford University Press.

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING '92 Proceedings of the 14th conference on Computational linguistics - Volume 2*

Joachims, T. (1999). Making large-scale SVM learning practical. In: B. Schölkopf, C. Burges, and A. Smola (Editors) *Advances in Kernel Methods - Support Vector Learning*, , MIT Press, 1999.

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics* , 33 (1), 159-174.

Lapata, M., & Keller, F. (2004). The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks. In: *Proceedings of HLT-NAACL'2004*

Lin, D., & Pantel, P. (2002). Concept Discovery from Text. In: *COLING '02 Proceedings of the 19th international conference on Computational linguistics - Volume 1*.

Maedche, A., & Staab, S. (2002). Measuring Similarity between Ontologies. *EKAW '02 Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. Springer-Verlag.

Maedche, A., & Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* , 16 (2), 72-79.

Maedche, A., & Volz, R. (2001). The Ontology Extraction Maintenance Framework Text-To-Onto *ICDM Workshop on integrating data mining and knowledge management*.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Medche, A. (2002). Ontology Learning for the Semantic Web (Vol. 665). Kluwer International.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM* , 38 (11), 39-41.

Morin, E. (1999). Extraction de liens sèmantiques entre termes à partir de corpus de textes techniques. Univesitè de Nantes, Facultè des Sciences et de Techniques.

Navigli, R., & Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Comput. Linguist.* , 30 (2), 151-179.

Padó, S. (2006). User's guide to sigf: Significance testing by approximate randomisation.

Pantel, P., & Pennacchiotti, M. (2006). Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: *ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Pekar, V., & Staab, S. (2002). Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In: *Proceedings of the Nineteenth Conference on Computational Linguistics* , 2, 786-792.

Penrose, R. (1955). A Generalized Inverse for Matrices. In: *Proc. Cambridge Phil. Soc. 51*

Ravichandran, D., & Hovy, E. (2002). Learning surface text patterns for a Question Answering System. In: *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*

Roark, B., & Bacchiani, M. (2003). Supervised and unsupervised PCFG adaptation to novel domains. In: *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* (p. 126-133). Association for Computational Linguistics.

Robison, H. R. (1970). Computer-detectable semantic structures. *Information Storage and Retrieval* , 6 (3), 273-288.

Scott, W. A. (1955). Reliability of Content Analysis: The Case of Nominal Scale Coding. *The Public Opinion Quarterly* , 19 (3), 321-325.

Snow, R., Jurafsky, D., & Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence., *ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (p. 801-808).

Szpektor, I., Tanev, H., Dagan, I., & Coppola, B. (2004). Scaling Web-based Acquisition of Entailment Relations. In: *Proceedings of EMNLP'2004*

Toumouth, A., Lehireche, A., Widdows, D., & Malki, M. (2006). Adapting WordNet to the Medical Domain using Lexicosyntactic Patterns in the Ohsumed Corpus. In: Proceedings of IEEE International Conference on. Computer Systems and Applications (p. 1029-1036). IEEE Computer Society.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer.

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In: *COLING '00 Proceedings of the 18th conference on Computational linguistics - Volume 2* (p. 947-953). Association for Computational Linguistics.

Yoshida, K., Tsuruoka, Y., Miyao, Y., & Tsujii, J. i. (2007). Ambiguous Part-of-Speech Tagging for Improving Accuracy and Domain Portability of Syntactic Parsers., In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (p. 1783-1788).

Zanzotto, F. M., Pennacchiotti, M., & Pazienza, M. T. (2006). Discovering Asymmetric Entailment Relations between Verbs Using Selectional Preferences. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (p. 849-856). Association for Computational Linguistics.

ADDITIONAL READING SECTION

KEY TERMS & DEFINITIONS

Ontology Learning: is a type of learning model where the learner (semi-)automatically extract relevant concepts and relations from a given corpus to create or to extend an ontology.

Probabilistic Model: is a type of learning model where the learner store probabilities or confidence weights in the model.

Logistic Regression: is a generalized linear model used for binomial regression

Pseudoinverse: is a generalization of the inverse matrix

Incremental Learning: is a type of learning model where the learner updates its model with new information

SVD:method for dimentionality reduction

Transitivity: relationship between three elements. If the relationship holds between the first and second elements and between the second and third elements, it necessarily holds between the first and third elements.

Semantic Turkey: is a platform for Semantic Bookmarking and Ontology Development

[i] The extensional definition of a concept is the enumeration of all its instances.

[ii] Considering “*dog*” as instance of “*animal*” is not completely correct as *dog* can be a concept in the structured knowledge repository. Yet, it is useful to describe the difference between *intensional* and *extensional* definitions.

[iii] We used the version 3.0 of WordNet