Dipartimento di Informatica Sistemi e Produzione

Università degli Studi di Roma Tor Vergata

Dottorato di Ricerca
Informatica e Ingegneria dell'Automazione
Ciclo XXII

Exploiting Transitivity in Probabilistic Models for Ontology Learning

Francesca Fallucchi

Supervisors:

Prof.ssa Maria Teresa Pazienza

Prof. Fabio Massimo Zanzotto

Rome, June 2010

To my family and
to the one which
will be it soon.

Abstract

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

We propose a novel probabilistic model for learning semantic networks that expands existing semantic networks taking into accounts both corpus-extracted evidences and the structure of the generated semantic networks. The model exploits structural properties of target relations such as transitivity during learning. The probability for a given relation instance to belong to the semantic networks of words depends both on its *direct* probability and on the *induced* probability derived from the structural properties of the target relation. Our model presents some innovations in estimating these probabilities.

We also propose a model that can be used in different specific knowledge domains with a small effort for its adaptation. In this approach a model is learned from a generic domain that can be exploited to extract new informations in a specific domain.

Finally, we propose an incremental ontology learning system: Semantic Turkey Ontology Learner (ST-OL). ST-OL addresses two principal issues. The first issue is an efficient way to interact with final users and, then, to put the final users decisions in the learning loop. We obtain this positive interaction using an ontology editor. The second issue is a probabilistic learning semantic networks of words model that exploits transitive relations for inducing better extraction models. ST-OL provides a graphical user interface and a human-computer interaction workflow supporting the incremental leaning loop of our learning semantic networks of words.

We empirically show that all our proposed models give a real contribute to the different considered tasks improving performance.

Contents

List of Tables

List of Figures

1

Introduction

Gottfried Wilhelm Leibniz, the famous German philosopher, was also a librarian. He was convinced that human knowledge was like a bazaar: a place full of all sorts of goods without any order or inventory. Searching specific knowledge items was then perceived to be a challenge. Nowadays, we have powerful machines to process and collect data that we can share using the World Wide Web. These technologies, combined with the human need of exchanging and sharing information, produced an incredibly large evolving collection of documents. The Web has become the modern worldwide scale knowledge bazaar where searching specific information is a titanic task. Ontologies represent the Semantic Web's reply to this need, providing shared metadata vocabularies for representing knowledge [Berners-Lee *et al.*(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. Thus, learning automatically these ontologies from document collections is the challenge in this context. This thesis wants to contribute in this area.

The rest of the Chapter is organized as follows. In Section 1.1 we give the definition of ontology in the Semantic Web and we describe its applications for organizing knowledge. In Section 1.2 we describe the methodologies for building ontologies and in general semantic networks from texts. In Section 1.3 we shortly introduce the contribution that this thesis gives to the state-of-the-art in learning knowledge from text. In the last Section the outline of the thesis is given (Section 1.4).

1.1 Ontologies and Knowledge Organization

Ontologies are one of the answers for organizing the modern worldwide scale knowledge bazaar provided in the context of the Semantic Web [Berners-Lee *et al.*(2001)]. The term ontology is borrowed from philosophy, where it is a systematic account of Existence. The Ontology with the capital O is only one. For knowledge-based systems, ontologies are an explicit, formal specification of a conceptualization needed to organize knowledge in specific domains and applications. These ontologies become then the reference point for applications because what “exists” is exactly what is represented [Gruber(1993)]. An ontology defines the basic terms, i.e., a sort of controlled vocabulary, and relations of a topic area as well as the rules for combining terms and the relations to define extensions to the vocabulary [Neches *et al.*(1991)].

In the Semantic Web vision, any service that wants to access and manipulate knowledge of a knowledge domain should use a shared ontological representation. As a benefit, these services will have the possibility to reason about the properties of that domain and to share knowledge with other services using the same or a related ontology to describe the domain. These new services are called "semantic services": services performing specific knowledge access and having their data (arguments as well as returned objects) explicitly described through ontology vocabularies. A "semantic service" is a wheel of the overall model for knowledge organization and sharing. In this context, shared ontologies improve relations between the actors, i.e., the services, involved in the overall model by helping in combining or replacing services that have a common semantic ground.

Nowadays, the Semantic web is spreading out. Ontologies are already used in databases design, knowledge management of organizations and companies, information seek on distributed databases, information extraction from texts, search engines, etc. In recent years many ontologies have been produced describing different knowledge domains: STEP (manufacturing), FOAF (Personal Information Management), UMLS (medical terminology), Gene Ontology (molecular biology), Agrovoc (agriculture), SWEET (Earth and Environmental Terminology), etc. Yet, the coverage of such ontologies with respect to the human knowledge is still low and efforts for increasing this coverage are needed.

1.2 Ontology engineering and ontology learning from text

Building, developing and managing ontologies is one of the major problems for making the Semantic Web vision possible. Ontology engineering is the field that studies the methods and the methodologies for finding solutions to the problem. There are three main approaches in ontology engineering:

- In the first approach, one or more people give a local and shared agreement on an ontology that meets his/her or their requirements. The obtained ontology is certainly accurate and fully meets the requirements but a shared agreement between people is anyway difficult to have. Furthermore, the ontology is not readily usable by other people or in application domains similar to what it was designed for.

- The second approach is the ontology development by reengineering of shared semi-structured knowledge as domain taxonomies or dictionaries. This approach leverages on reuses something that is already shared. Then, the final ontological resource should be ready to be commonly agreed. This model has two main limitations: the first is that it could not map perfectly the requirements of the specific domain; the second is that the reused shared knowledge may not be understood by others.

- The third approach is to adapt existing general ontological components to target domains. This help in linking new ontological content with already shared components. Unfortunately, there are no tools that allow to immediately reuse existing ontology components.

In all the above approaches ontologies are manually created and then the task is time consuming and expensive. Automatically learning ontologies from domain textual collections can help in solving this last issue as well as the above ones. Automatically extracting ontological knowledge from domain document collections can speed up the production and help in determining shared knowledge representations. Domain document collections represent what domain experts think about concepts and concept relations within the domain. Retrieving and structuring knowledge using these collections can help in finding shared ontological representations. It is worth noticing that ontology learning methods extract knowledge from texts where concepts are described with words. When extracting relations among concepts, these models ultimately learn relations among words.

Automatically learning ontological information from documents is still far from being an accomplished goal. Current methods are often referred as semi-automatic ontology learning and suggest possible ontologies that should be later controlled. This thesis gives a contribution in this area.

1.3 Thesis Contributions

In this thesis, we will firstly consider ontologies as ultimately used in language-based applications. Then, we will focus our attention to semantic networks of words. Our main goal is to contribute practically to the research on semantic networks learning models by covering different aspects of the task.

We propose a novel probabilistic model for learning semantic networks that expands existing semantic networks taking into accounts both corpus-extracted evidences and the structure of the generated semantic networks. The model exploits structural properties of target relations such as transitivity during learning. The probability for a given relation instance to belong to the semantic networks of words depends both on its *direct* probability and on the *induced* probability derived from the structural properties of the target relation. Our model presents some innovations in estimating these probabilities.

To estimate *direct* probability, we naturally exploit vector space reduction techniques for selecting features, i.e. we leverage on the computation of logistic regression based on the Moore-Penrose pseudo-inverse matrix to exploit unsupervised feature selection by singular value decomposition (SVD).

To estimate *induced* probability, we directly include transitivity in the formulation of the probabilistic model. This *induced* probability should capture the fact that a decision on the pair (i, j) depends also on the transitive relations activated by (i, j). In the *induced model*, we exploit *direct* probabilities to derive the *induced* probability. We introduce three models for exploiting the probabilistic definitions of concepts within the *induced* probabilistic model: *intensional*, *extensional* and *mixed* model.

Then, we propose a model that can be used in different specific knowledge domains with a small effort for its adaptation. In this approach a model is learned from a generic domain that can be exploited to extract new informations in a specific domain. To evaluate the new specific domain networks of words we consult human annotators and we examine inter-annotator agreement.

Finally, we present an incremental ontology learning system: Semantic Turkey Ontology Learner (ST-OL). ST-OL addresses two principal issues. The first issue is an efficient way to interact with final users and, then, to put the final users decisions in the learning loop. We obtain this positive interaction using an ontology editor. The second issue is a probabilistic learning semantic networks of words model that exploits transitive relations for inducing better extraction models. ST-OL provides a graphical user interface and a human-computer interaction workflow supporting the incremental leaning loop of our learning semantic networks of words .

1.4 Thesis Outline

In Chapter 2 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 Chapter 3 and 4 we introduce our two probabilistic models to learn semantic networks of words. The first, described in Chapter 3, estimates direct probability between words using our logistic regressors based on the Moore-Penrose pseudo inverse matrix. With the second model, described in Chapter 4, we exploit structural properties of target relations in determining the probability of the word pairs to be in a particular relation. We show, in particular, three different ways to exploit transitivity.

In Chapter 5 we propose a semantic networks learning method that can exploit models learned from a generic domain to extract new information in a specific domain.

In Chapter 6 we present Semantic Turkey Ontology Learner (ST-OL), 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.

Finally, in Chapter 7 we draw some final conclusions and we outline feature research directions.

2

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 thesis wants to give a contribution.

In the following we analyze these techniques thoroughly, with particular reference to aspects and components that characterize them, limitations included. In Section 2.1 we describe the layer cake organization of the task for ontology learning from text [Buitelaar *et al.*(2005)]. The overall task is seen as composed by different subtasks for learning terms, concepts, taxonomies, relations among concepts, and axioms/rules. In Section 2.2 we focus on the learning of relations among concepts/words by introducing the three classical working hypotheses. We will analyze the feature spaces built from corpus-extracted evidences using these three hypotheses. In Section 2.3 we describe how semantic networks learning methods use the three hypotheses and exploit relation properties such as transitivity. In Section 2.4 we focus on a particular model that uses a probabilistic formulation for semantic networks learning. In Section 2.5 we introduce a model to learn semantic networks in new domains and in Section 2.6 we present an incremental learning model. Finally, in Section 2.7 we introduce feature selection models to solve the problem of the huge feature spaces arising in semantic networks learning models.

2.1 Learning Ontologies from Text

Ontology learning was originally started in [Maedche and Staab(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.

For devising the problem, the complex task of ontology learning has been divided in subtasks organized in a layer cake [Buitelaar *et al.*(2005)]. Figure

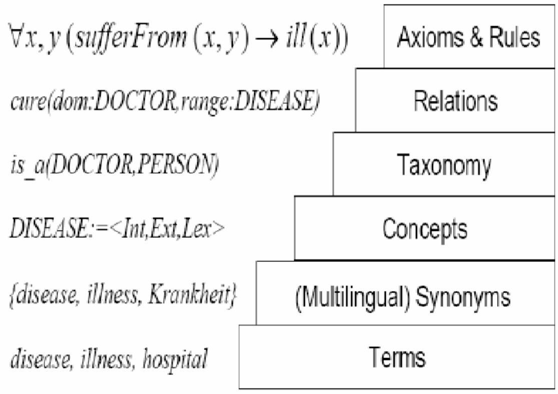

∀x, y (sufferFrom (x, y) → ill (x))
Axioms & Rules
cure(dom:DOCTOR,range:DISEASE)
Relations
is_a(DOCTOR,PERSON)
Taxonomy
DISEASE:=<Int,Ext,Lex>
Concepts
{disease, illness, Krankheit}
(Multilingual) Synonyms
disease, illness, hospital
Terms

Figure 2.1: Ontology Learning layer cake

2.1 shows this layer cake along with an example. The basic level learns terms of the domain, e.g., *disease*. At the synonymy level, terms are grouped in synonymy sets representing concepts of the domain, e.g., {*disease*, *illness*}. At the concept level, concepts of the domain are generated along with their intensional and extensional definitions and their lexicalization. At the taxonomical level, generalization relations among concepts are retrieved and modeled, e.g., the taxonomic relation of the concept *doctor* with the concept *person*. At the relation level, all other relations are considered, e.g., the *cure* relation among *doctor* and *disease*. Finally, at the level of rules and axioms, methods should derive rules such as the one reported in Figure 2.1.

Upper levels in the layer cake generally correspond to a higher complexity of the algorithms. In the following we will illustrate all the levels of the layer cake, from the bottom to the top (see Figure 2.1).

Terms

At the first level there is the terms extraction, where terms are linguistic realizations of domain-specific concepts. In this subtask methods provide relevant terms for the construction of ontology concepts. Typically, these systems are established by integrating components for the parsing and statistical components. Terms extraction is based on information retrieval methods for term indexing. Text indexing systems based on the assignment of appropriately weighted single terms produce retrieval results that are superior to those obtainable with other more elaborate text representations [Salton and Buckley(1987)]. The statistical components determine the degree of association between the words in the term. For example in [Pantel and Lin(2001)] an algorithm is given that use a language independent statistical corpus-based to extract terms. As reported in [Bourigault *et al.*(2001)], new insights on computational terminology can be found in articles about automatic analysis, storage, and use of terminology and applied in linguistics, computational linguistics, information retrieval, and artificial intelligence. At the present, researchers are interested in developing a domain independent method for the automatic extraction of multi-word terms from machine-readable special language corpora. This method should combine linguistic and statistical information [Frantzi *et al.*(2000)].

Synonyms

In the second level there are the methodologies for the recognition of synonyms, in view of identifying a set of lexical variants in which the same concept can be expressed and then identified. These techniques exploits Harris' hypothesis, i.e.

words are semantically similar if they share linguistic contexts [Harris(1968)]. There are several synonym acquisition methods based on Harris' distribution hypothesis. For example UNICON [Lin and Pantel(2001b)], that is an unsupervised algorithm, can be used for inducing a set of concepts, each one consisting of a cluster of words and a set of concepts that may be constructed for any corpus. Clustering By Committee (CBC) [Lin and Pantel(2002)], is a clustering algorithm that automatically discovers concepts from text. It initially discovers a set of tight clusters called committees that are well scattered in the similarity space (the centroid of the members of a committee is used as the feature vector of the cluster). It then proceeds by assigning elements to their most similar cluster. In [Hindle(1990)], Hindle describes a method of determining the similarity of nouns on the basis of a metric derived from the distribution of subject, verb and object in a large text corpus. The resulting quasi-semantic classification of nouns obtained with this method demonstrates the plausibility of the distributional hypothesis with potential applications in a variety of tasks, such as automatic indexing, resolving nominal compounds, and determining the scope of modification. Important techniques for synonym discovery belong to the family of latent semantic indexing algorithms [Landauer and Dumais(1997)] and other variants. In very specific domains, acquisition of synonyms has exploited approaches to word sense disambiguation such as in [Navigli and Velardi(2004), Turcato *et al.*(2000), Buitelaar and Sacaleanu(2002)]. Currently, in acquisition of synonyms, the statistical information measures defined over the web seem to be a trend [Baroni and Bisi(2004), Turney(2001)].

Concepts

In the third level there is the formulation of concepts. Techniques adopted for learning synonyms and concepts strongly overlap because both subtasks typically use the distributional hypothesis.

Unfortunately, in the extraction of concepts from text it is not clear what exactly constitutes a concept. For this reason concept formulation should provide an intensional and extensional definition of the concept and a set of linguistic relations among the concepts.

In the intensional definition of the concept we give a (in)formal definition of the set of objects that the concept describes. For example, *a disease is an impairment of health or a condition of abnormal functioning*, Figure 2.1. In the extensional definition of the concept a set of "instances" is given that is described by the definition of the concept. For example, for the concept *disease* the possible instances are *influenza, cancer, heart disease.* In the linguistic relations we consider the term itself and its multilingual synonyms. Respect to the considered example, we have *influenza, cancer, heart disease, etc.*

In [Navigli and Velardi(2004)] Navigli and Velardi have developed a system (OntoLearn) to learn concepts intentionally. With this system the association of a complex concept to a complex term is provided using a semantic interpretation based on the structural semantic interconnections, i.e. a word sense disambiguation algorithm. In this way, WordNet is trimmed and enriched with the detected domain concepts.

Many other researchers are working in the extensional definition of concepts. For example in [Evans(2003)] hierarchies of named entities are derived from text and concepts are discovered from an extensional point of view. The concepts as well as their extensions are thus derived automatically. In Know-It-All system [Etzioni *et al.*(2004)] the extensions of existing concepts are learned and the existing concepts are populated with instances. The main limitations of these approaches are that they require large training data that usually must be manually built.

Unfortunately, the language may consist of strings describing the intuitive meaning of a concept in natural language such as for the *glosses* of the WordNet, where the word form is distinguished from its meaning by introducing the so called *synsets*, i.e. words sharing a common meaning in some context.

Taxonomy

Nowadays, the problems put in evidences for the first levels can be solved by efficient existing technologies. On the contrary, the higher levels still present unsolved acquisition problems that we want to address in this thesis with innovative methods. For this reason in this thesis we propose methods to solve the acquisition problems of these higher levels. The approaches that get the best results in the taxonomy level are based on integrated approaches that use both corpus-extracted evidences and existing language resources such as WordNet [Basili *et al.*(2007)]. A large variety of methods have been proposed to learn taxonomies. Usually, these learning methods are based on three working hypotheses: Basic Hypothesis (BH), Harris' Distributional Hypothesis (DH), and Lexico-Syntactic Pattern exploitation hypothesis (LSP). These working hypotheses are largely described in Section 2.2.

BH relies on the analysis of co-occurrence of terms in the same sentence, for example, automatically deriving a hierarchical organization of concepts from a set of documents looking only to the concepts and the external knowledge repositories such as WordNet [Miller(1995)]. For example, it possible to measure the semantic similarity or relatedness between a pair of concepts (or word senses) using WordNet such as in [Pedersen *et al.*(2004)]. DH is widely used in many approaches for taxonomy induction from texts. For example, it is used for populating lattices, i.e. graphs of a particular class of formal concepts [Cimiano *et al.*(2005)]. Other researchers have mainly exploited hierarchical clustering algorithms to automatically derive taxonomies [Cimiano *et al.*(2004)]. Learning methods based on distributional hypothesis can be applied only for learning cotopy [Harris(1964), Deerwester *et al.*(1990)] and generalization [Geffet and Dagan(2005), Cimiano *et al.*(2005)]. Lexico-syntactic patterns (LSP) models are more general. In fact, these models can be potentially used for deciding whether or not any type of semantic relation holds between two words. This approach has been widely used for detecting hyperonymy relations [Hearst(1992a), Morin(1999), Snow *et al.*(2006)], but also for other ontological relations [Pantel and Pennacchiotti(2006)], even more relations [Szpektor *et al.*(2004), Ravichandran and Hovy(2002)], and for relations among verbs [Zanzotto *et al.*(2006), Chklovski and Pantel(2004)]. These learning models generally use the hypothesis that two words have a particular relation if they frequently appear in specific text fragments.

All the above mentioned models operate on concepts and not directly on words. But semantic relations are ultimately binary relations among words unless Word Sense Disambiguation (WSD) systems are used. For example, the lexical substitution task in [McCarthy and Navigli(2007)] is accomplished using a model that operates on words. When applied as networks of words, taxonomies are tested in text based tasks, e.g. substituting words with similar ones in a given specific context. In this way, the task of learning or expanding these networks using plain text collections is feasible.

Relations

Several models have been elaborated to discoverer relations from text. In [Pantel and Pennacchiotti(2006), Snow *et al.*(2006)] generic semantic relations between concepts are learned using a model that operate as binary classifiers. In this case the task is deciding whether or not two words are in a specific semantic relation. Lexico-syntactic patterns are used as features to build vector spaces for word pairs where binary classifiers are applied. Feature values describe the correlation between contexts of word pairs and specific patterns. This approach is extremely relevant because the task is seen as a simple binary classification problem and not as a more complex multi-classification task [Pekar and Staab(2002)].

In [Maedche and Staab(2000)] an algorithm is proposed to discover conceptual relations building on shallow text processing techniques. It is a generalized association rule algorithm that does not only detect relations between concepts, but uses term co-occurrence to determine the appropriate level of abstraction at which the relations are defined. In [Hearst(1992a), Szpektor *et al.*(2004), Pantel and Pennacchiotti(2006)] are determined relations among relevant words using the textual patterns. Generic semantic relations between nouns have been extracted exploiting lexical-syntactic patterns in [Pantel and Pennacchiotti(2006), Szpektor *et al.*(2004)]. The same approach has been used to find specific relations between verbs in [Chklovski and Pantel(2004), Zanzotto *et al.*(2006)].

Whenever these models deal with relations among synonymy sets or concepts. The approach proves to be limited, because these relations are usually hard to be retrieved. In fact, these models can produce semantic relations among concepts only under particular conditions, i.e. if used with specific text collection where concepts are generally expressed with stable terms, learning models produce semantic relations among concepts [Navigli and Velardi(2004), Medche(2002), Cimiano *et al.*(2005)]. What we will exploit in this thesis is that the natural product of these methods is the semantic relations among words and not among concepts.

Axioms & Rules

The last level is expected to develop systems that infer rules and axioms from texts. This field has been poorly explored, but it is achieving growing attention in literature. Many researchers are deriving lexical entailment rules. The main focus hereby is to learn lexical entailments for application in question answering systems such as in PASCAL Network of Excellence Recognizing Textual Entailment (RTE) Challenge [Dagan and Glickman(2005)]. The RTE task is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred (entailed) from another one. This application independent task is suggested as capturing major inferences about the variability of semantic expressions which are commonly needed across multiple applications.

2.2 Three working hypotheses for knowledge learning from text

Semantic resources are ultimately exploited in text understanding systems as networks of words. Here, we want to review the three basic hypotheses for extracting relations among concepts: Basic Hypothesis (BH), Harris' Distributional Hypothesis (DH), and Lexico-Syntactic Pattern exploitation hypothesis (LSP). These hypotheses are used in any semantic networks learning methods are based. The three hypotheses clearly define different spaces where *target textual forms* and *contexts* may be represented. These spaces are, respectively, the *space of the target textual forms* and the *space of the contexts.* These spaces are strictly interconnected and what is done in one space can be exploited in the other. Figure 2.2 shows the two spaces and presents an example that can clarify how the three hypotheses work together.

2.2.1 The Basic Hypothesis

The well-known *Basic Hypothesis* (BH) is applied when the relation or the similarity between the textual forms, w_i and w_j, is determined looking only at w_i and w_j and at external knowledge repositories such as WordNet [Miller(1995)]. The similarity or oriented relation between textual forms may be defined as function $r_{BH}(w_i, w_j)$ operating only in the space of the target textual forms. For example, $r_{BH}(w_i, w_j)$ may detect the similarity between w_i and w_j, where

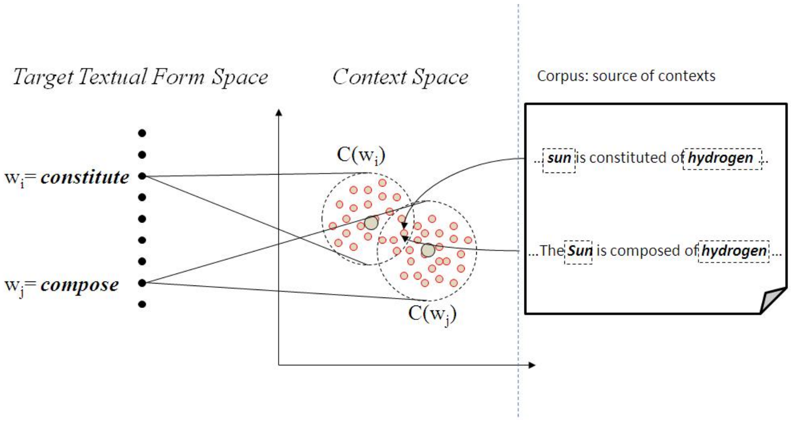

Target Textual Form Space
Context Space
Corpus: source of contexts
w_i= constitute
w_j= compose
C(w_i)
C(w_j)
...sun is constituted of hydrogen ...
...The Sun is composed of hydrogen ...

Figure 2.2: Two spaces for the three hypotheses

w_i and w_j are lemmas, using similarity measures defined over WordNet, as those collected in [Pedersen *et al.*(2004)]. Contexts are not used. This r_{BH} can then detect the similarity between *compose* and *constitute* (Figure 2.2), i.e., $r_{BH}(compose, constitute)$, looking at the two words and at an external resource only. More complex ways of computing the similarity using the basic hypothesis have been proposed when the forms are word sequences such as terms (e.g. [Jacquemin(2001)]) or complete sentences (e.g. [Dolan *et al.*(2004), Burger and Ferro(2005)]).

2.2.2 The Distributional Hypothesis

The well-known *Distributional Hypothesis* (DH) [Harris(1964)] allows determining whether or not two forms are in relation looking at their contexts. These latter are found in a corpus. The hypothesis states that *two forms are similar* *if these are found in similar contexts*, i.e., $r_{DH}(w_i, w_j) \approx sim_{DH}(C(w_i), C(w_j))$ where $C(w_i)$ and $C(w_j)$ are the contexts of the forms w_i and w_j in a given corpus. In the example of Figure 2.2, using the distributional hypothesis, the similarity between *compose* and *constitute* is determined as the overlap between $C(\mathit{constitute})$ and $C(\mathit{compose})$. As consequence, $sim_{DH}(C(w_i), C(w_j))$ is high. The words *compose* and *constitute* are similar because they can be found in similar contexts such as *the sun is constituted of hydrogen* and *the sun is composed of hydrogen*, i.e., the contexts containing both *sun* and *oxygen*. The $sim_{DH}(C(w_i), C(w_j))$ similarity is defined in the space of the contexts.

2.2.3 The Lexico-Syntactic Pattern Exploitation Hypothesis

Lexico-syntactic patterns exploitation hypothesis (LSP) used in [Robison(1970)] and used afterwards in [Hearst(1992a), Pantel and Pennacchiotti(2006), Szpektor *et al.*(2004)], allows to determine relations among relevant words using textual patterns, e.g. X *is constituted of* Y may be used to determine the *part-of* relation among X and Y. The textual patterns are defined in the textual form space and, then, are used in the context space to retrieve textual elements in relation. In the above example of Figure 2.2, the pattern X *is constituted of* Y is used to find the *part-of* relation between the two words *sun* and *oxygen*. More generally, $C(\mathit{constitute})$ contains, after statistical filtering, words that are in a *part-of* relation. Moreover, the equivalences determined in the textual form space, e.g., the equivalence between *constitute* and *compose*, can be used to further augment words that are in a given relation. Using the equivalence between *constitute* and *compose*, elements in *part-of* relation can be found in $C(constitute) \cup C(compose)$.

LSP learning models generally use the hypothesis that two words have a particular relation if they frequently appear in specific text fragments. Prototypical text fragments related to a particular relation are often called lexico-syntactic patterns. For example, given the *isa* relation, X *isa* Y if X and Y are frequently found in contexts such as *"X is a Y"*, *"X as well as Y"*, or *"X, Y,"*. Given the relation R, a pair of words (X, Y), and the patterns related to the relation R, the above mentioned learning methods tend to determine a confidence weight that expresses to which degree the relation R holds for the pair (X, Y) according to a collection of documents.

2.3 Semantic network learning methods

Models for automatically learning semantic networks of words from texts use both corpus-extracted evidences and existing language resources [Basili *et al.*(2007)]. In the previous sections we have define the three hypotheses underlying any method for learning relations among concepts. In this section we focus on how existing resources are used and we see as the existing learning models do not explicitly exploit structural properties of target relations when learning taxonomies or semantic networks of words.

How existing resources are used is an important aspect in this task. DH models generally start learning from scratch. In [Cimiano *et al.*(2005)], for example, lattices and related semantic networks are built from scratch. Even when prior knowledge is used in DH models [Pekar and 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. LSP models (e.g., [Snow *et al.*(2006)]) offer a more uniform way to represent prior and extracted knowledge. In this case, the insertion of a new word in the hierarchy is seen as a binary classification problem.

In the rest of the section we will see the approaches to induce isa-relation (Section 2.3.1) and other kind of relations from text (Section 2.3.2). Finally the limits of these approaches will be reported in Section 2.3.3.

2.3.1 Learning isa relation

The distributional hypothesis is widely used in many approaches for semantic networks induction from texts. As defined in Section 2.2.2 the distributional hypothesis (DH) states that *words occurring in the same contexts tend to be similar*. Then, it can be naturally applied to determine relatedness or sisterhood between words. 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 *et al.*(2005)] for populating lattices (i.e. graphs of a particular class) of formal concepts. Namely, the DH is exploited to extract attributes for objects. Nodes of the lattice are obtained clustering objects with similar attributes and hierarchical links are drawn between two nodes A and B if the set of attributes of A are an included subset of attributes of B. These lattices are then used to build taxonomic hierarchies. The idea of drawing semantic networks links using the inclusion of features derived exploiting the distributional hypothesis has been also used in [Geffet and Dagan(2005)] where the *distributional inclusion hypothesis* is defined.

The *distributional inclusion hypothesis* [Geffet and Dagan(2005)] and the similar formal concept analysis [Cimiano *et al.*(2005)] basically state that the word "*a*" is the generalization of the word "*b*" if the properties representing the contexts of "*a*" are included in those representing the contexts of "*b*". Semantic network learning methods based on these hypothesis can be applied only for learning cotopy, for example sisterhood in generalization networks [Harris(1964), Deerwester *et al.*(1990)] and generalization [Geffet and Dagan(2005), Cimiano *et al.*(2005)].

2.3.2 Learning all other relations

For all the other kind of relations, the basic working hypothesis is the exploitation of LSP's ([Robison(1970)]), i.e., as already described in Section 2.2.3, a generic way to express a semantic relation in texts. These models have been applied for learning is-a relations [Hearst(1992a), Snow *et al.*(2006)], generic semantic relations between nouns [Pantel and Pennacchiotti(2006), Szpektor *et al.*(2004)], and specific relations between verbs [Zanzotto *et al.*(2006), Chklovski and Pantel(2004)]. 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.

LSP models can be potentially used for deciding whether or not any type of semantic relation holds between two words. These models have been widely used for detecting hyperonymy relations [Hearst(1992a), Morin(1999)], and also for other ontological relations [Pantel and Pennacchiotti(2006)], even more generic [Szpektor *et al.*(2004), Ravichandran and Hovy(2002)], and for relations among verbs [Zanzotto *et al.*(2006), Chklovski and Pantel(2004)]. 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(1992a), Morin(1999), Pantel and Pennacchiotti(2006), Chklovski and Pantel(2004), Ravichandran and Hovy(2002), Zanzotto *et al.*(2006), Szpektor *et al.*(2004)]

- these models coherently exploit existing taxonomies in the expansion phase

[Snow *et al.*(2006)]

- the classification is binary, i.e., a word pair belongs or not to the taxonomy [Pantel and Pennacchiotti(2006), Snow *et al.*(2006)]

In this way, a single classifier is associated to each treated relation. In this thesis, we will select a probabilistic approach, among LSP semantic networks learning models, because in this way we can model both existing and new knowledge with probabilities. This is needed to positively exploit transitivity during learning.

2.3.3 Limits

LSP models are interesting because they can learn any kind of semantic relations but they do not explicitly exploit structural properties of target relations when learning taxonomies or semantic networks of words. Semantic relation learning models based on the distributional hypothesis 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.

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 *et al.*(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.

2.4 Using Probabilistic Models

Semantic network learning approach exploit three working hypotheses to build feature space. These feature spaces are used to determine whether or not new pairs of words coming from the text collection have to be included in existing knowledge repositories. With the approach proposed by Snow in [Snow *et al.*(2006)] it is possible to take into account both corpus-extracted evidences and existing language resources using a probabilistic formulation. This model is the only one using, even if only intrinsically, one of the properties of semantic networks (transitivity) to expand existing networks.

In this section, we will firstly motivate why we should store probabilities or confidence weights in learnt semantic networks (Section 2.4.1). Then, we will introduce the state-of-the-art of probabilistic semantic networks learning models (Section 2.4.2) and, finally, we will point out their limits (Section 2.4.3).

2.4.1 Confidence weights, probabilities, and corpus-based learning

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., [Lin and Pantel(2001a), Snow *et al.*(2006), Pantel and Pennacchiotti(2006)]). 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 *et al.*(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.

Ambiguity of natural language is the second reason why knowledge repositories should store confidence weights (or probabilities) of extracted knowledge items. 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 *et al.*(2004)]. Experience in different NLP tasks such as part-of-speech (POS) tagging suggests that it is important to model and store these probabilities. In [Yoshida *et al.*(2007)] a comparison between three POS taggers is shown: one emitting one interpretation per word, one emitting multiple interpretations, and, finally, one emitting multiple interpretations with associated probabilities. The POS taggers have been then evaluated with respect to the performances obtained by a parser. Even if the probabilistic model of the parser is different with respect to the one of the POS tagger, the parser has better performances with the third POS tagger that emits tags and the associated probabilities.

2.4.2 Iterative probabilistic model

We illustrate here an existing state-of-the-art probabilistic model for semantic networks learning presented in [Snow *et al.*(2006)]. This probabilistic model is the only one based on lexico-syntactic patterns that intrinsically uses transitivity to expand existing semantic networks. We will hereafter call this model *iterative*. We are interested in this model because it represents a valid alternative to the models proposed in this thesis. The iterative model, instead of determining induced probabilities, iteratively adds facts in the knowledge base changing the initial semantic networks.

The task of learning semantic networks from a corpus is seen as a probability maximization problem. The semantic networks are seen as a set T of assertions R over pairs $R_{i,j}$. If $R_{i,j}$ is in T, i is a concept and j is one of its generalizations. For example, $R_{dog,animal} \in T$ describes that *dog* is an *animal*. The main innovation of this probabilistic method is the ability of taking into account in a single probability the information coming from the corpus and an existing semantic networks T.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \in T)$ of an assertion $R_{i,j}$ to belong to the semantic network T and (2) the posterior probability $P(R_{i,j} \in T|\overrightarrow{e}_{i,j})$ of an assertion $R_{i,j}$ to belong to the semantic network T given a set of evidences $\overrightarrow{e}_{i,j}$ derived from the corpus. The evidences are 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"*. These, among many other features, are indicators of an is-a relation between i and j (see [Hearst(1992a)]).

Given a set of evidences E over all the relevant word pairs, the probabilistic semantic networks learning task is defined as the problem of finding a semantic network $\widehat{T}$ that maximizes the probability of observing the evidences E, i.e.:

$$\widehat{T} = \arg\max_{T} P(E|T)$$

This maximization problem is solved by a local and iterative search. Each step maximizes the ratio between the likelihood $P(E|T')$ and the likelihood $P(E|T)$ where $T' = T \cup I(R_{i,j})$ and $I(R_{i,j})$ are the added relations. This ratio is called multiplicative change $\Delta(N)$ and is defined as follows:

$$\Delta(I(R_{i,j})) = P(E|T')/P(E|T) \tag{2.1}$$

Given the semantic network T and the relation $R_{i,j}$, the set $I(R_{i,j})$ contains $R_{i,k}$ if $R_{j,k}$ is in T and contains $R_{k,j}$ if $R_{k,i}$ is in T. For example: given T and $R_{dog,animal}$, if $R_{animal,organism} \in T$ then $I(R_{dog,animal})$ contains $R_{dog,organism}$.

Moreover, given T and $R_{bird,beast}$, if $R_{turkey,beast} \in T$ then $I(R_{bird,beast})$ contains $R_{turkey,beast}$.

The main innovation of this model is the possibility of adding at each step the best relation $\{R_{i,j}\}$ as well as all the relations induced from $R_{i,j}$ and the existing semantic networks T. Two different approaches can be adopted at this point:

flat: at each iteration step, a single relation is added,

$$\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(R_{i,j})$$

inductive: at each iteration step, a set of relations $I(\widehat{R}_{i,j})$ is added

$$\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(I(R_{i,j}))$$

Finally, it is possible to demonstrate that the following equation holds:

$$\Delta(R_{i,j}) \quad = \quad k \cdot \frac{P(R_{i,j} \in T|\vec{e}_{i,j})}{1 - P(R_{i,j} \in T|\vec{e}_{i,j})} \qquad (2.2)$$

where k is a constant that will be neglected in the maximization process. The model for predicting $P(R_{i,j} \in T|\vec{e}_{i,j})$ is then trained using logistic regression.

2.4.3 Limits

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)]. But even large knowledge repositories such as WordNet [Miller(1995)] are extremely poor when used in specific domains (e.g., medicine [Toumouth *et al.*(2006)]).

Obtaining manually structured knowledge repositories in specific domains is a very time consuming and expensive task. 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 to expand, we will not have enough data for training the initial model. In general, despite the scarcity of domains covered by existing structured knowledge, there are no limits on the domains where these resources may be required to operate. In general, in learning methods the amount of out-of-domain data is larger than in-domain data. For this reason, we will 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 be liked a learning semantic networks of words model that can be used, with a small effort for the adaptation, in different specific knowledge domains.

2.5 Adapting semantic networks to new 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(1992a), Pantel and Pennacchiotti(2006), Snow *et al.*(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 *et al.*(2006), Fallucchi and Zanzotto(2009b)]. 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 *et al.*(2006)], all WordNet has been used as source of training examples. In this case, domain adaptation techniques must be adopted [Bacchiani *et al.*(2004), Roark and Bacchiani(2003), Chelba and Acero(2006), Daumé and Marcu(2006), Gao(2009), Gildea(2001)].

Domain adaptation is a well-known problem in machine learning and statistical learning. To stress the difference between the distribution of the data in the original domain (also called *background domain*) and in the target domain, we can refer to *out-of-domain* data and as *in-domain* data. *Out-of-domain* data are generally large sets and are used for training. In general, in learning method the assumption that out-of-domain data and in-domain data share the same underlying probability distribution is inaccurate. This problem arises for many applications. Generally, in-domain data are drawn from a distribution that is related, but not identical, to out-of-domain distributions of the training data. Some the amount of out-of-domain data is generally larger than in-domain data, we need to envisage methods that exploit these data for building accurate in-domain models.

The domain adaptation problem exactly consists in leveraging out-of-domain data to derive models well performing on in-domain data. The alternative is a manually building initial training resources for new domains. But this is an expensive task just as designing a system for each target domain. The natural expectation from the domain adaptation models is to minimize the efforts required to build in-domain data using a model trained with out-of-domain data. This context, it becomes very important to adapt existing models from rich source domains to resource poor target domains. The problem of domain adaptation arises in a large variety of applications: natural language processing [Blitzer *et al.*(2006), Chelba and Acero(2006), Daumé and Marcu(2006)], machine translation [Bertoldi and Federico(2009)], word sense disambiguation [Chan and Ng(2007)], etc..

In the rest of this section we will investigate the domain adaptation techniques (Section 2.5.1), the model adaptation (Section 2.5.2) and the limits of the presented approaches (Section 2.5.3).

2.5.1 Domain Adaptation Techniques

Different domain adaptation techniques are introduced in the context of specific applications and statistical learning methods. A standard technique used in statistical language modeling and in other generative models is the maximum a posteriori (MAP) estimation [Gauvain and Lee(1994)], where the prior knowledge is used to estimate the model parameters. In the MAP estimation, the some model parameters are considered to be random variables with a known distribution (the prior one). Then, the prior distribution and the maximum likelihood distribution based on the in-domain observations are used to derive the posterior distribution of the parameters, from which the model is selected. If the amount of in-domain data is large, the mode of the posterior distribution is mostly defined by the adaptation sample; if the amount of adaptation data is small, the mode will nearly coincide with the mode of the prior distribution.

The intuition behind the MAP estimation is that, once there are sufficient observations, the prior model need no longer to be relied upon corpus, and which are more general. The MAP framework is general enough to include some previous model adaptation approaches, such as corpus mixing [Gildea(2001)]. The MAP estimation has been also used in [Roark and Bacchiani(2003)] to adapt a lexicalized probabilistic context-free grammar (PCFG) to a novel domain. In [Chelba and Acero(2006)] a MAP adaptation technique for maximum entropy models has been developed for the problem of recovering the correct capitalization of uniformly case text for language modeling in speech recognition. In [Daumé and Marcu(2006)] a statistical formulation has been provided, that is a mixture of the maximum entropy model and the linear Chain Models for conditional random fields.

Two other classes of model adaptation methods are very interesting: error-driven learning approaches and model interpolation approaches. In the error-driven learning approaches, the background model is adjusted to minimize the ranking errors made by the model on the adaptation data [Bacchiani *et al.*(2004), Gao(2009)]. In the model interpolation approaches, the in-domain data are used to derive an adaptation model, which is then combined with the background model trained on the out-of-domain data. In [Gao(2009)] the model interpolation has been investigated for web search ranking.

2.5.2 Model Adaptation

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. In [Gao(2009)] a set of error-driven learning methods is developed where, in an incremental way, each feature weight could be changed separately but also new features could be constructed.

In [Blitzer *et al.*(2006)] a common representation is given for features extracted from different domains. Pivot features from unlabeled data are used to put domain-specific words in correspondence where the pivot features are features occuring frequently in the two domains and behaving similarly in both. By analogy with [Blitzer *et al.*(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.

In some cases, many steps may be required to adapt a model trained on the source domain to the target domain [Roark and Bacchiani(2003), Ando(2004), Daumé and Marcu(2006)]. On the contrary, in the approach that we propose we learn a model from the out-of-domain data that can be used to learn the in-domain data without any additional effort.

2.5.3 Limits

In general, in the learning method, the amount of out-of-domain data is larger than in-domain data. For this reason, we want to envisage methods that exploit out-of-domain data for building accurate in-domain models. 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.

2.6 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 and Volz(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 *et al.*(2004)] is a Protégé [Gennari *et al.*(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 and Manzano-Macho(2003)] of the OntoWeb project. A more recent example is offered by the Text2Onto

[Cimiano and Volker(2005)] plug-in for the Neon toolkit [Haase *et al.*(2008)], a renewed version of Text-To-Onto with improvements featuring ont-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 *et al.*(2007)] the authors define a web browser extension based on the Semantic Turkey Knowledge Acquisition Framework [Griesi *et al.*(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 and 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.

2.7 Feature selection models

Knowledge harvesting and semantic networks learning models exploit the three working hypotheses to build feature spaces where instances (words as in [Pekar and Staab(2002)] or word pairs as in [Snow *et al.*(2006)]) are represented. Decision models are learned using existing knowledge repositories and then applied to new words or word pairs. Generally, all learning models use as features all the possible and relevant generalized contexts where words or word pairs can appear. For example, possible features in the word pair classification problem are *"is a"* and *"as well as"*. These feature spaces can be huge, as they include all potential relevant features for a particular relation among words, where relevant features are not known in advance. Yet, large feature spaces can have negative effects on machine learning models such as increasing the computational load and introducing redundant or noisy features. Feature selection can solve these collateral problems (see [Guyon and Elisseeff(2003)]).

Feature selection is a process wherein a subset of the features available from the data is selected for application in a learning algorithm. The best subset contains the least number of dimensions that most contribute to accuracy. Feature selection models are also widely used in semantic networks learning methods. For example, attribute selection for building lattices of concepts in [Cimiano *et al.*(2005)] is done applying specific thresholds on specific information measures on attribute extracted from corpora. This model uses conditional probabilities, point-wise mutual information, and a selectional preference-like measure ([Resnik(1993)]). The wide range of feature selection models can be classified in two main families: *supervised* feature selection models (Section 2.7.1) and *unsupervised* feature selection models (Section 2.7.2).

2.7.1 Supervised Feature selection models

Supervised models directly exploit the class of training instances for determining weather a feature is relevant or not. The idea is to select features that are highly correlated with final target classes. The most standard filtering features can be estimated in different ways, such as with information theoretic ranking criteria, according to their individual predictive power. In particular, when adopting the information theoretic ranking criteria, mutual information and information gain are often used [Dhillon *et al.*(2003)].

Intuitively, mutual information is a measure of the amount of information that one random variable contains about the other. The higher is the value the less is the uncertainty of one random variable due to knowledge about the other. In this type of supervised feature selection the question is to find optimal word clusters in terms of preserving mutual information between words.

The information gain is the reduction in the uncertainty about a word when we know the other one. The uncertainty about a word is measured by its entropy. Information gain is primarily used in decision tree algorithm.

2.7.2 Unsupervised Feature selection models

Unsupervised models are used when the classification of training instances is not available at the training time or it is inapplicable, such as in information retrieval. Straightforward and simple models for unsupervised feature selection can be derived from information retrieval weighting schemes, an example is the term frequency times inverse document frequency ($tf * idf$) reported in Section 2.7.2.1. In machine learning, a mathematical construct is largely used as feature selection that directly reveals the rank and the corresponding ideal basis of a dataset: singular value decomposition (SVD) reported in Section 2.7.2.2. We will describe SVD with much detail because this technique has been successfully integrated in the probabilistic models that we suggest in this thesis.

2.7.2.1 Term frequency times inverse document frequency

Term frequency times inverse document frequency, $tf * idf$, is a statistical measure that combines term frequency (tf) and inverse document frequency (idf). The first, tf, allows weighting the relevance of a term in a corpus in a simple way. The measuring of the term relevance is by the absolute term frequency, i.e. the number of times a term occurs in a corpus. While the second, idf, penalizes terms which occur in several documents. Then, $tf * idf$ allow weighting how important a word is for a document in a collection or corpus. In $tf * idf$, relevant features are respectively those appearing more often or those being more selective, i.e., appearing in fewer instances.

2.7.2.2 Singular Value Decomposition

In machine learning, singular value decomposition (a mathematical construct that directly reveals the rank and the corresponding ideal basis of a dataset) is largely used as unsupervised feature selection.

SVD is one of the possible factorizations of a rectangular matrix that has been largely used in information retrieval for reducing the dimension of the document vector space [Deerwester *et al.*(1990)]. The decomposition can be defined as follows. Given a generic rectangular $n \times m$ matrix A, its singular value decomposition is:

$$A = U \Sigma V^T$$

where U is a matrix $n \times r$, V^T is a $r \times m$ and Σ is a diagonal matrix $r \times r$. The two matrices U and V are unitary, i.e., $U^T U = I$ and $V^T V = I$. The diagonal elements of Σ are the *singular values* $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$ where r is the rank of the matrix A. For the decomposition, SVD exploits the linear combination of rows and columns of A.

A first trivial way of using SVD as unsupervised feature reduction is the following. Given E as a set of training examples represented in a feature space of n features, we can represent it as a matrix, i.e. a sequence of examples $E = (\overrightarrow{e_1}...\overrightarrow{e_m})$. With SVD, the $n \times m$ matrix E can be factorized as $E = U\Sigma V^T$. This factorization implies that we can focus the learning problem on a new space using the transformation provided by the matrix U. This new space is represented by the matrix:

$$E' = U^T E = \Sigma V^T \tag{2.3}$$

where each example is represented with r new features. Each new feature is obtained as a linear combination of the original features, i.e. each feature vector $\overrightarrow{e_l}$ can be seen as a new feature vector $\overrightarrow{e_l}' = U^T\overrightarrow{e_l}$. When the target feature space is big and the cardinality of the training set is small, i.e., $n >> m$, the application of SVD results in a reduction of the original feature space dimension to the rank r of the matrix E is $r \leq min(n, m)$.

A more interesting way of using SVD as unsupervised feature selection model is to exploit its approximated computations, i.e. :

$$A \approx A_k = U_{m\times k}\Sigma_{k\times k}V^T_{k\times n} \tag{2.4}$$

where k is smaller than the rank r. The computation algorithm [Golub and Kahan(1965)] allowed to stop at a given k different from the real rank r. The property of the singular values, i.e., $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$, guarantees that the first k are bigger than the discarded ones. There is a direct relation between the informativeness of the dimension and the value of the singular value. High singular values correspond to dimensions of the new space where examples have more variability whereas low singular values determine dimensions where examples have a smaller variability (see [Liu(2007)]). These dimensions can not be used as discriminative features in learning algorithms. The possibility of computing the approximated version of the matrix gives a powerful method for feature selection and filtering as we can decide in advance how many features or, better, linear combination of original features we want to use.

SVD is an unsupervised feature selection model in the sense that the feature selection is done without taking into account the final classes of the training examples. This is not always the case: feature selection models, such as those based on Information Gain, largely use the final classes of training examples. SVD as feature selection is independent of the classification problem.

3

Approaches to estimate direct probabilities

Any corpus-based knowledge learning method augments existing knowledge repositories with new information extracted from texts. Two big issues must be addressed in this process. The first is that we are mixing reliable with unreliable information. The second is that, as we are dealing with natural language, an ambiguity affects the bits of the discovered knowledge. This is the reason why knowledge repositories should store confidence weights or probabilities of the extracted knowledge items. Here, we will estimate this probability, which will be hereafter called *direct* probability, it being obtained directly from the observations over text collection.

In this chapter, we will firstly define the *direct* probabilistic model (Section

3.1) and we will introduce the logistic regression model (Section 3.2). Then, we will show how regression coefficients are estimated (Section 3.3) and we will describe how SVD can be used as feature selector in the logistic regression that estimates the probabilities of the model (Section 3.4). Finally, we will report and we comment the results of the experiments (Section 3.5) and we will draw some conclusions (Section 3.6).

3.1 Direct Probabilistic Model

The *direct probabilistic model* is directly built on the observations over the text collection. We model the semantic network learning problem as a binary classification task, in line with [Pantel and Pennacchiotti(2006)] and [Snow *et al.*(2006)]. Given a pair of words (i, j) and a vector of observed features $\vec{e}_{i,j}$, we want to build a binary classifier that determines if "i" is in a relation R with "j" and gives the related confidence weight.

In the *direct probabilistic model*, we define 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:

$$P(R_{i,j} \in T|E) \tag{3.1}$$

where E is a set of evidences extracted from the corpus. We will hereafter refer to this probability as $P(R_{i,j}|E)$.

Using the assumption of independence of the evidence vectors, we can rewrite equation (3.1) as $P(R_{i,j}|\vec{e}_{i,j})$ where $\vec{e}_{i,j}$ is the set of evidences for (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 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 j ([Hearst(1992a)]). The *direct* probabilities $P(R_{i,j}|\vec{e}_{i,j})$ only depend on what has been observed in the corpus for a particular pair of words (i, j).

The last issue we need to address is how to estimate the *direct* probabilities $P(R_{i,j}|\vec{e}_{i,j})$ using an initial knowledge base and a corpus where evidences for pairs (i, j) can be extracted. We will do this using the logistic regression model [Cox(1958)] that, as we will see, gives a natural setting for using singular value decomposition (SVD) as unsupervised feature selection model.

3.2 Logistic Regression

Logistic Regression [Cox(1958)] is a particular type of statistical model for relating responses Y to linear combinations of predictor variables X. It is a specific kind of Generalized Linear Model (see [Nelder and Wedderburn(1972)]) where the function is the *logit function* and the dependent variable Y is a *binary* or *dicothomic* variable which has a Bernoulli distribution. The dependent variable

Y takes value 0 or 1. The probability that Y has value 1 is a function of the regressors $x = (1, x_1, ..., x_k)$.

The *direct* probability $P(R_{i,j}|\vec{e}_{i,j})$ falls in the category of the probabilistic models where the logistic regression can be applied, because $R_{i,j} \in T$ can be seen as the binary dependent variable and $\vec{e}_{i,j}$ as the vector of its regressors. We start from describing formally the logistic regression model. Given a binary stochastic variable Y and a generic stochastic variable X for the regressors, we can define p as the probability of Y to be 1 given $X = \vec{x}$, i.e.:

$$p = P(Y = 1|X = \vec{x})$$

The distribution of Y is a Bernoulli distribution.

Given the definition of the $logit(p)$ as:

$$logit(p) = \ln\left(\frac{p}{1-p}\right) \tag{3.2}$$

and given the fact that Y is a Bernoulli distribution, the logistic regression predicts that the logit is a linear combination of the values of the regressors, i.e.,

$$logit(p) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \tag{3.3}$$

where $\beta_0, \beta_1, ..., \beta_k$ are called *regression coefficients* of the variables $x_1, ..., x_k$ respectively.

Given the regression coefficients, it is possible to compute the probability of a given event where we observe the regressors x to be $Y = 1$ or in our case to belong to the taxonomy. This probability can be computed as follows:

$$p(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}$$

Probability	*Odds*	*Logit*
$0 \leq p < 0.5$	$[0, 1)$	$(-\infty, 0]$
$0.5 < p \leq 1$	$[1, \infty)$	$[0, \infty)$

Table 3.1: Relationship between probability, odds and logit

It is trivial to determine the $odds(R_{i,j})$ related to the multiplicative change of the probabilistic taxonomy model. The *odds* is the ratio between positive and negative events. It is defined as follows:

$$odds(R_{i,j}) = \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1-P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} \tag{3.4}$$

It is noteworthy that the odds is strictly related to the logit

$$odds(R_{i,j}) = \exp(\beta_0 + \overrightarrow{e}_{i,j}^{T}\beta) \tag{3.5}$$

The relationship among probability, odds and logit is show in Table 3.1.

3.3 Estimating Regression Coefficients

The last issue is how to estimate the regression coefficients. This estimation can be done using the maximal likelihood estimation. The above *logit* definition generate a set of linear equations. The linear problem is then solved introducing a pseudo-inverse matrix, the original matrix being usually rectangular and singular.

The importance of obtaining the regression coefficients (see above) is that we can estimate in this way a probability $P(R_{i,j}|\vec{e}_{i,j})$ given any configuration of the regressors $\vec{e}_{i,j}$, i.e., the observed values of the features.

The estimation of the β coeffients can be done as follows. Let assume to have a multiset O of observations extracted from a corpus. Elements of the multiset are $(y, \vec{e}_{i,j})$ where $y = 1$ if (i, j) is a positive case and $y = 0$ if (i, j) is a negative case. We can now generate a set E of all the different vectors $\vec{e}_{i,j}$. For the sake of simplicity, we will write $\vec{q}$ instead of $\vec{e}_{i,j}$. For each $\vec{q} \in E$, we can use the maximum likelihood to estimate the initial probability $P(Y = 1|\vec{q})$ as the frequency of the pair $(1, \vec{q})$ in O divided by the frequency of $\vec{q}$. For each $\vec{q} \in E$, we have then a set of equations of this kind:

$$logit(P(Y = 1|\vec{q})) = \beta_0 + \beta_1 q_1 + ... + \beta_m q_m \tag{3.6}$$

where m is the size of the feature space. This set of equations can be written as a linear equation system:

$$\overrightarrow{logit(p)} = Q\beta \tag{3.7}$$

where Q is a matrix that includes a constant column of 1's. The matrix is:

$$Q = \begin{pmatrix} 1 & q_{11} & q_{12} & \cdots & q_{1m} \\ 1 & q_{21} & q_{22} & \cdots & q_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & q_{n1} & q_{n2} & \cdots & q_{nm} \end{pmatrix}$$

The set of equations in (3.7) is a particular case of multiple linear regression [Caron *et al.*(1988)].

As Q is a rectangular and singular matrix, Q is not invertible and the system $\overrightarrow{logit(p)} = Q\beta$ has no solutions. Yet, it is possible to use the principle of the

Least Square Estimation. With this principle we can determine the solution β that minimize the residual norm, i.e.:

$$\widehat{\beta} = \arg\min \|Q\beta - \overrightarrow{logit(p)}\|^2 \tag{3.8}$$

This problem can be solved by the **Moore-Penrose pseudoinverse** Q^+ [Penrose(1955)] that gives the following final equation:

$$\widehat{\beta} = Q^+\overrightarrow{logit(p)} \tag{3.9}$$

It is important to remark that if the inverse matrix exists then $Q^+ = Q^{-1}$ and Q^+Q , QQ^+ are symmetric.

3.4 Computing Pseudoinverse Matrix with SVD Analysis

We can finally illustrate why it is natural to use singular value decomposition as feature selection in a probabilistic taxonomy learner. In the previous sections we described how the probabilities of the taxonomy learner can be estimated using logistic regressions and we concluded that a way to determine the regression coefficients β is by computing the **Moore-Penrose pseudoinverse** Q^+. Here we compute **Moore-Penrose pseudoinverse** using the SVD [Penrose(1955)]. Given a SVD of the matrix $Q = U\Sigma V^T$, the pseudo-inverse matrix that minimizes the Equation 3.8 is:

$$Q^+ = V\Sigma^+U^T \tag{3.10}$$

The diagonal matrix Σ^+ is the $r \times r$ transposed matrix of Σ having as diagonal elements the reciprocals of the Σ singular values $\frac{1}{\delta_1}, \frac{1}{\delta_2}, ..., \frac{1}{\delta_r}$.

The use of SVD in the computation of the pseudo-inverse matrix intrinsically designate SVD as unsupervised feature selection model. We can compute different approximations of the pseudo-inverse matrix. The algorithm for computing the singular value decomposition is iterative [Golub and Kahan(1965)]. The firstly derived dimensions are those with higher singular value. Once obtained the higher value we can decide how many dimensions we want to use, considering that the first k dimensions are more informative than the $k+1$ dimensions. We can take different values of k in order to obtain different SVD as approximations Q_k^+ of the original matrix Q^+ (Equation 3.10)

$$Q_k^+ = V_{n\times k} \Sigma_{k\times k}^+ U_{k\times m}^T$$

where Q_k^+ is a matrix n by m obtained taking the first k singular values.

The computation algorithm [Golub and Kahan(1965)] can be stopped at a given k's different from the real rank r. The property of the singular values, i.e., $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$, guarantees that the first k are bigger than the discarded ones.

A direct relation holds between the informativeness of the i-th new dimension and the singular value δ_i. High singular values correspond to dimensions of the new space where examples have more variability whereas low singular values correspond to dimensions where examples have a smaller variability [Liu(2007)]. The latter dimensions can be then hardly used as efficient features selection in learning.

The computation of approximated matrices is a powerful method for feature selection and filtering as we can decide in advance how many features or, better, linear combinations of the original features we want to use.

3.5 Experimental Evaluation

In this section, we will show two sets of experiments where the validity of our *direct* probabilistic model has been checked. With the first set (Section 3.5.2) we will determine if keeping probabilities within the final knowledge base is better than taking strict decisions. To assess this claim we will compare our *direct* probabilistic model with the **iterative model** presented in Section 2.4.2 that is the state-of-the-art of the probabilistic models. With the second set (Section 3.5.1) we will empirically explore whether the use of SVD feature selection positively affects performances of the probabilistic semantic networks learner.

3.5.1 Experimental Set-up: keeping probabilities within the final knowledge base

Here, we report and discuss the first set of experiments to determine if keeping probabilities within the final knowledge base is better than taking strict decisions. To assess this claim we compare our *direct* probabilistic model with the **iterative model** . To this aim an appropriate experimental setting has been disposed (corpus, semantic networks and feature spaces).

3.5.1.1 Corpus

The adapted corpus is the *English Web as Corpus* (ukWaC) [Baroni *et al.*(2009)]. This is a web extracted corpus of about 2700000 web pages containing more than 2 billion words. The corpus contains documents of 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 and Keller(2004)].

3.5.1.2 Semantic Networks

As target semantic networks we selected a portion of WordNet[1] [Miller(1995)]. Namely, we started from 44 concrete nouns divided in 3 classes: animal, artifact, and vegetable. For the sake of comprehension, this set is reported in Table 3.2.

For each word w, we selected the synset s_w that generalizes the class it belongs to. In this way we obtained a set S of synsets. We then expanded the set to S' adding the siblings (i.e., the coordinate terms) for each synset in S. The sets S' contains 265 coordinate terms plus the 44 original concrete nouns.

For each element in S we collected the hypernyms, obtaining the set H of the hypernyms. We then removed from the set H the top classes *entity*, *unit*, *object*, and *whole*, obtaining 77 hypernyms. For the purpose of the experiments we derived a taxonomy T from S and S' and a taxonomy $\overline{T}$ from the set of negative examples. The taxonomy T is the portion of WordNet implied by $O = H \cup S'$, i.e., T contains all the $(s, h) \in O \times O$ that are in WordNet.

On the contrary, $\overline{T}$ contains all the $(s, h) \in O \times O$ that are not in WordNet.

[1] We used the version 3.0 of WordNet

	Concrete nouns	*Clas*	*Sense*		*Concrete nouns*	*Clas*	*Sense*
1	banana	Vegetable	1	23	boat	Artifact	0
2	bottle	Artifact	0	24	bowl	Artifact	0
3	car	Artifact	0	25	cat	Animal	0
4	cherry	Vegetable	2	26	chicken	Animal	1
5	chisel	Artifact	0	27	corn	Vegetable	2
6	cow	Animal	0	28	cup	Artifact	0
7	dog	Animal	0	29	duck	Animal	0
8	eagle	Animal	0	30	elephant	Animal	0
9	hammer	Artifact	1	31	helicopter	Artifact	0
10	kettle	Artifact	0	32	knife	Artifact	0
11	lettuce	Vegetable	2	33	lion	Animal	0
12	motorcycle	Artifact	0	34	mushroom	Vegetable	4
13	onion	Vegetable	2	35	owl	Animal	0
14	peacock	Animal	1	36	pear	Vegetable	0
15	pen	Artifact	0	37	pencil	Artifact	0
16	penguin	Animal	0	38	pig	Animal	0
17	pineapple	Vegetable	1	39	potato	Vegetable	2
18	rocket	Artifact	0	40	scissors	Artifact	0
19	screwdriver	Artifact	0	41	ship	Artifact	0
20	snail	Animal	0	42	spoon	Artifact	0
21	swan	Animal	0	43	telephone	Artifact	1
22	truck	Artifact	0	44	turtle	Animal	1

Table 3.2: Concrete nouns, Classes and senses selected in WordNet

We then have 4596 positive pairs in T and 48354 negative pairs in $\overline{T}$. To obtain the training and testing sets, we randomly divided the set $T \cup \overline{T}$ in two parts, $T_{tr} \cup \overline{T}_{tr}$ and $T_{ts} \cup \overline{T}_{ts}$, respectively the 70% and 30% of the original $T \cup \overline{T}$.

3.5.1.3 Feature Space

We used a bag-of-n-gram feature space for implicitly modeling lexical-syntactic patterns. Features are words (monograms), bigrams, and trigrams. The n-grams represent specific lexico-syntactic patterns. Given a pair $(i, j) \in T \cup \overline{T}$, we built the related feature vector $\vec{e}_{i,j}$ using the contexts where the words i and j appears in a window of 5 words at most. For each context of (i, j), the word sequence between i and j was used to increment the frequency of the related n-gram. For example, given the pair $(car, vehicle)$, we can retrieve the context:

... to control the ***car*** *as a motor* ***vehicle*** *and ...*

where the only features that can be obtained from the word sequence between the two target words are: *as*, *a*, *motor*, *as a*, *a motor*, and *as a motor*.

3.5.1.4 Results

In the experiment we have analyzed the relevance of the probability in the final knowledge base. We evaluated the *iterative* and the *direct* probabilistic models on their ability of sorting the pairs. The *iterative* model adds some pairs at each step. On the contrary, the *direct* probabilistic model, produces a sorting of the pairs according to the probabilities. We then compared the two methods. In the case of the *iterative* methods, we have plotted the curve that relates the accuracy to the number of added pairs. The accuracy is computed as the number

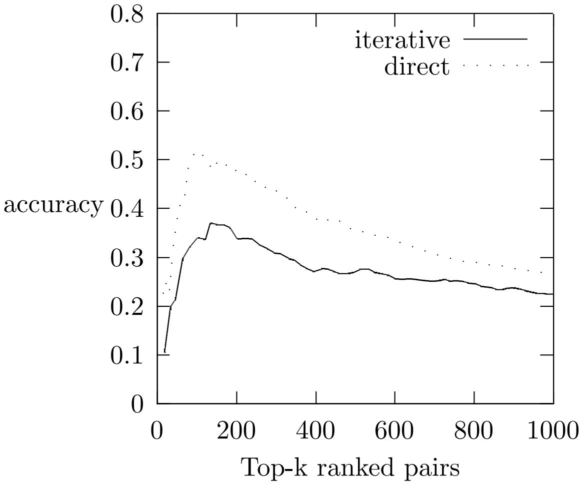


Figure 3.1: Accuracy of the top-k ranked pairs for the iterative and direct probabilistic semantic networks learners

of correctly added pairs with respect to the added pairs. On the contrary, in the case of the probabilistic model we have plotted the accuracies with respect to the ranked pairs. For this set of experiments, the SVD approximated pseudo-inverse matrix was computed with $k = 100$.

The results are reported in Figure 3.1. We can observe that, keeping the probabilities is better than making a decision at each step, i.e. the *direct* model outperforms the *iterative* model. Thus, the final knowledge base should keep the probabilities. The accuracies are reported in Table 3.3 for both models for two different cuts of the sorted pair list. The second and the third columns report, respectively, the accuracies for 100 and 1000 considered pairs.

	Top k-pairs	
Probabilistic Model	*100*	*1000*
iterative	0.33981	0.22500
direct	0.51000	0.26200

Table 3.3: Accuracy of the different models at top 100 and 1000 ranked pairs

3.5.2 Experimental Set-up: feature selections affect performances

In this section, we empirically explore whether the use of SVD feature selection positively affects performances of the probabilistic semantic networks learner. In the experiments, we addressed two issues:

- determining to what extent SVD feature selection affects performances of the semantic networks learner;
- determining if, for the probabilistic semantic networks learner, SVD is better than other simpler models for supervised and unsupervised feature selection.

We explore the effects on both the **flat** and the **inductive** probabilistic semantic networks learner. In the **flat** model the best relation is added at each iteration step, whereas in **inductive** model not only the best relation but even all the relations entailed by it are added at each iteration step (Section 2.4.2).

3.5.2.1 Corpus

The adapted corpus was the *English Web as Corpus* (ukWaC) [Baroni *et al.*(2009)] that we described in Section 3.5.1.1 .

3.5.2.2 Semantic Networks

As target semantic networks we selected a portion of WordNet[2] [Miller(1995)] as we described in Section 3.5.1.2.

3.5.2.3 Feature Space

The focus of the experiments is the analysis of the effect of the SVD feature selection. We used both n-grams and bag-of-words as feature spaces. Out of the $T \cup \overline{T}$, we selected only those pairs appearing at a distance of 3 tokens at most. Using these 3 tokens, we generated three spaces: (1) the 1-gram space that contains monograms, (2) the 2-gram space that contains monograms and bigrams, and (3) the 3-gram space that contains monograms, bigrams, and trigrams. For the purpose of this experiment, we used a reduced stop list as classical stop words because punctuation, parenthesis and the verb *to be* are very relevant in the context of features for learning a semantic networks.

The *baseline model* only contains Heart's patterns [Hearst(1992a)] as features. The feature value is the point-wise mutual information. These features are in some sense the best features for the task as these have been manually selected after a process of corpus analysis. We included in our 3-gram model

[2] We used the version 3.0 of WordNet

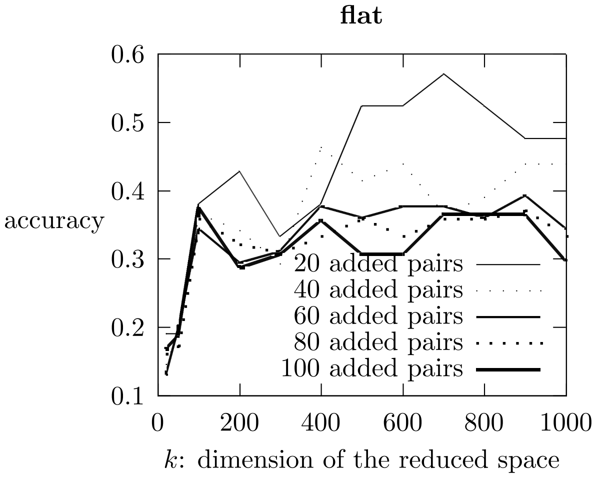
also the features obtained with the *baseline model*, in order to compare our best models with manual selected features.

3.5.2.4 Feature Selection

We want define the feature selection models we compared against. As *unsupervised feature selection models* we used the term frequency times the inverse document frequency (*tf*idf*). Instances $\overrightarrow{e}$ have the role of the documents. As *supervised feature selection models* we used the mutual information (mi). For all the feature selection models, we selected the first k's features. Finally, we used a manual feature selection model based on the Heart's patterns [Hearst(1992b)] that we called *manual model*. In this model we used only classical Hearst's patterns as features. We proposed two experimental settings: a *natural* and an *artificial* one. In the *natural setting* we used only positive pairs for the training set, that is the natural situation when augmenting existing taxonomies because only positive word pairs can be derived from existing taxonomies. In the *artificial setting* we used both positive and negative examples. We empirically explored with three set of experiments, whether the use of the SVD feature selection positively affects the performances of the probabilistic semantic networks learner.

3.5.2.5 Results

With the first set of experiments we focused on the attention whether or not performances of the probabilistic taxonomy learner is positively affected by the proposed feature selection model based on the singular value decomposition.



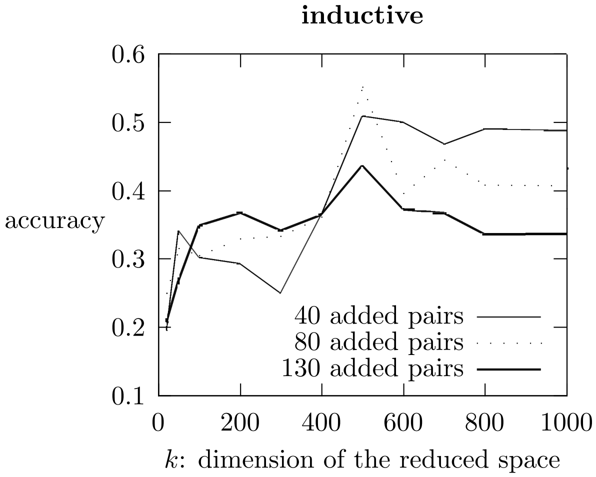


Figure 3.2: Accuracy over different cuts of the feature space

We then determined the performance with respect to different values of k, when k represents the number of surviving dimensions where the pseudo-inverse is computed and thus, the number of features adopted in the model. We performed this first set of experiments in the 1-gram feature space. Punctuation was considered.

Figure 3.2 reports the accuracy of the probabilistic learner with respect to the size of the feature set, i.e. the number k of single values considered for computing the pseudo-inverse matrix. We reported these curves even for different sizes of the set of added pairs to determine if the effect of the feature selection is preserved during the iteration of the local search algorithm. Curves were reported for both the *flat* and the *inductive* model, the *flat* algorithm adds one pair at each iteration. We reported the curves for every 20 added pairs. Each curve shows that accuracy does not increase after a dimension of k=700. This size of the space is necessary only for the first 20 added pairs. Accuracy increases up to k=700 and then decreases. When we add more pairs, the optimal size of the space is around k=200. For the *inductive* model we report the accuracies for 40, 80, 130 added pairs. Here, at each iteration, more than one pair is added. The optimal dimension of the feature space seems to be k=500, the performances decreasing or staying stable after this value. SVD feature selection has then a positive effect for both the *flat* and the *inductive* probabilistic semantic networks learners. This has beneficial effects both on the performances and on the computation time.

With the second set of experiments we determined whether or not SVD feature selection for the probabilistic taxonomy learner behaves better than a

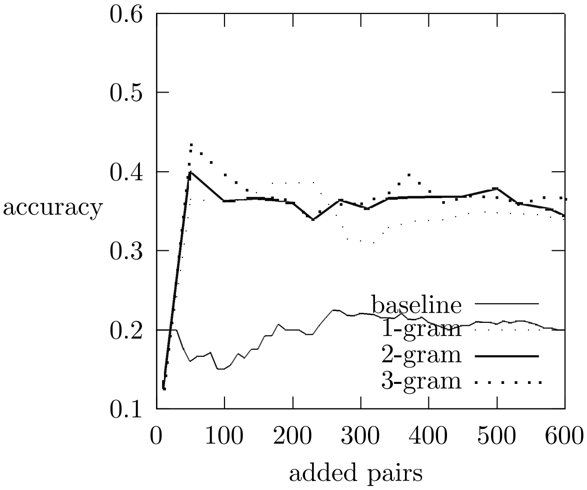
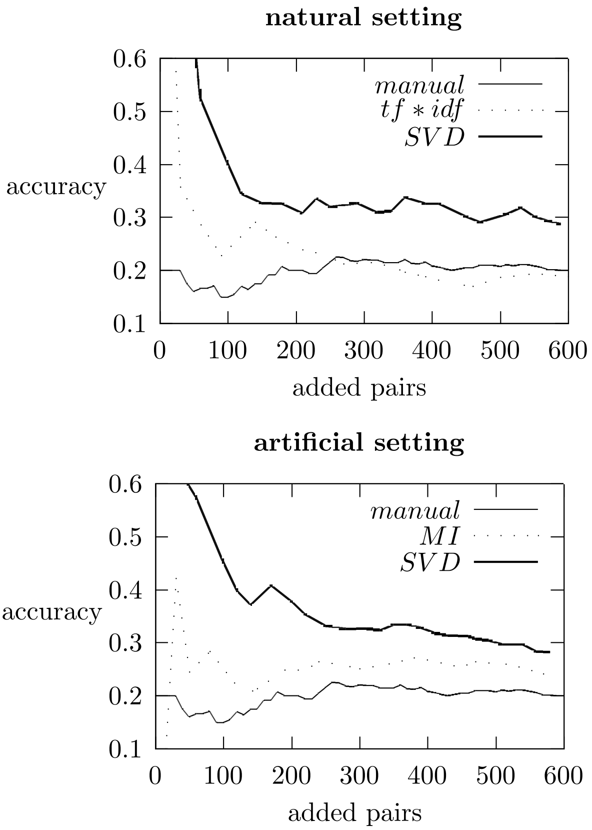


Figure 3.3: Comparison of different feature spaces with k=400

reduced set of known features. We fixed the dimension k to 400 and we compared the *baseline model* with different probabilistic models with different feature sets: 1-gram, 2-gram, and 3-gram. We can consider that the trigram model before the cut on its dimensions contains feature subsuming the *baseline model*. The results are reported in Figure 3.3.

The curves report the accuracy after n added pairs. All the probabilistic models outperform the baseline model. As in the case of the first series of experiments (see Figure 3.2) more informative spaces such as 3-gram behaves better when the number of added pairs is small. Performances of the three reduced pairs become similar after 100 added pairs. These experiments show that SVD feature selection has a positive effect on performances as resulting



natural setting

artificial setting

Figure 3.4: Comparison of different feature selection models

models are always better with respect to the baseline.

With the last set of experiments we determined whether or not SVD feature selection for the probabilistic taxonomy learner behaves better than other feature selection models. We fixed k to 600 both for the SVD selection model and for the other feature selection models. In these experiments, the original feature space is the bigram space.

The results are reported in Figure 3.4. The curves report the accuracies of the different models after n added pairs. In the *natural setting*, we compared our model against the $tf * idf$ and the *manual feature selection* and we deduced that our SVD model outperforms both feature selection models. The same for the mutual information (MI) in the *artificial setting.* Our SVD way of selecting features proved to be very effective.

3.6 Conclusion

We presented a model to naturally introduce SVD feature selection in a probabilistic semantic networks learner. The method is effective as allows the designing of better probabilistic semantic networks of words learners.

The *direct* model proved to outperform the state-of-the-art of the semantic networks learner models.

It is important also to note that our SVD-based logistic approach demonstrates here all its efficiency. The iterative model, described in Section 2.4.2, requires the computation of the regression at each step. On the contrary the most expensive part of the computation of the regression model we proposed (i.e. the computation of the pseudo-inverse matrix) is computed only once for all the iterative process. In Equation 3.9, the estimated $\widehat{\beta}$ changes at each step because the estimated $\overrightarrow{logit(p)}$ changes. The use of other regression methods such as Support Vector Machines [Cortes and Vapnik(1995)] is computationally unfeasible because they require to recompute the regression at each step.

4

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 *et al.*(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 chapter is organized as follows. In Section 4.1, we informally introduce our probabilistic model that explicitly used transitivity in semantic networks learning models. In Section 4.2, we formalize the probabilistic definitions of concepts in an *induced* probabilistic model. In Section 4.3, we propose three different methods for modeling induced probabilities. Finally, in Section 4.4, we want to demonstrate that our *induced* models can effectively exploit transitivity when we replicate an existing networks or we expand or build new semantic networks.

4.1 Probabilistic definitions of concepts in semantic networks learning

In this section, we want to informally introduce our inductive probabilistic model for semantic networks learning [Fallucchi and Zanzotto(2010)]. We have seen why we should store probabilities or confidence weights in learnt semantic networks of words in Section 2.4.1. In Section 4.1.1 we expanded these reasons including semantic relations with structural properties. We will then introduce our idea for giving *probabilistic definitions of concepts* that allows building our probabilistic model for semantic networks learning (Section 4.1.2).

4.1.1 Probabilities in semantic relations with structural properties

We have seen why we should store probabilities or confidence weights in learnt semantic networks of words in Section 2.4.1. 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 and 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 n 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 4.1(a)), using transitivity we can derive the *induced* relation, i.e., *dog* is an *animal* (dashed arrow in Figure 4.1(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 4.1(b)), using the transitivity we can derive the *induced* relation, i.e., *canine* is an *animal* (dashed arrow in Figure 4.1(b)). Another example is shown in Figure 4.1(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 *et al.*(2006)], these reliable and unreliable information is mixed during the knowledge acquisition process. In these models, if "*canine*" is an "*animal*" (see Figure 4.1(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*" hs been neglected.

4.1.2 Probabilistic definitions for concepts

Keeping and propagating uncertainty in transitive semantic networks is extremely important. We thus propose an *inductive semantic networks learning model*, i.e., a probabilistic semantic networks learning model based on lexico-syntactic patterns that exploits transitivity during learning and for determining

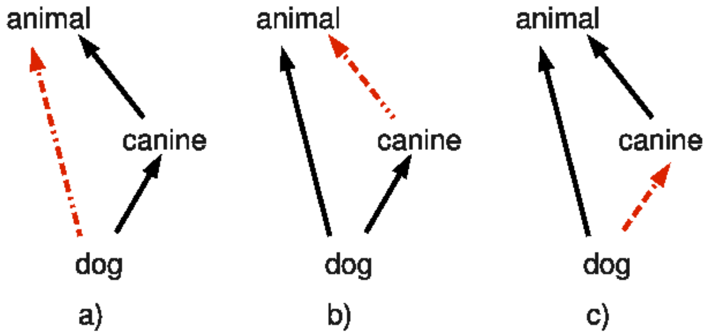


Figure 4.1: Examples of relations derived exploiting the transitivity

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 a "*animal*" with specific features. In the case of "*animal*", this relation contributes, in a wide sense, to the *extensional* definition[1] of "*animal*". It is like we are giving one of the possible instances [2] of the concept "*animal*". These formal *intensional* and

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

[2] 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

extensional definitions are often used to derive the similarity among words or concepts. *Cotopy* [Maedche and Staab(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 4.1, 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, defined in Section 3.1, and *induced probabilities* the second one.

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 model respectively: the *intensional inductive probabilistic model*, the *extensional probabilistic inductive model* and the *mixed probabilistic inductive model.* To give an intuitive idea of our models, we can use the example in Figure 4.1.

The *intensional inductive model* exploits direct *intensional* definitions to derive an induced *intensional* definition. In Figure 4.1.(a), we have as direct

between *intensional* and *extensional* definitions.

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 4.1.(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 4.1.(c)). Using all the other relations, we can derive also the induced probability of the relation "*dog*" is a "*canine*".

4.2 Inductive Probabilistic Model

In this section, we formalize the probabilistic definitions of concepts in an *induced* probabilistic model. In Section 4.3, 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

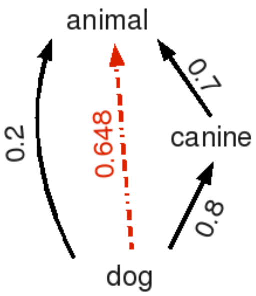


Figure 4.2: Example of relations derived exploiting transitivity

semantic relation.

As in [Pantel and Pennacchiotti(2006), Snow *et al.*(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 $\vec{e}_{i,j}$, we want to build a binary classifier that determines if i is a j and gives the related confidence weight. As in [Snow *et al.*(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.*

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 4.2), 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 4.2 we have the following *direct* probabilities (where $d = dog$, $a = animal$, and $c = canine$): $P(R_{d,a}|\vec{e}_{d,a}) = 0.2$, $P(R_{d,c}|\vec{e}_{d,c}) = 0.8$, and $P(R_{c,a}|\vec{e}_{c,a}) = 0.7$.

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 $\widehat{R}_{i,j}$ and the related probability function:

$$P(\widehat{R}_{i,j} \in T|E) \tag{4.1}$$

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 $P(\widehat{R}_{i,j}|E)$ 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 4.2. The *induced intensional* probability $P(\widehat{R}_{d,a}|E)$ can be computed as the probability of the event $\widehat{R}_{d,a} = R_{d,a} \cup (R_{d,c} \cap R_{c,a})$. This captures that the *induced* event $\widehat{R}_{d,a}$ 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:

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

Given this initial idea, we formalize our *induced* probabilistic models in the next sections.

4.3 Three inductive probabilistic models

We propose three different methods for modeling induced probabilities. We call these *intensional* (Sec.4.3.1), *extensional* (Section 4.3.2), and *mixed* model (Section 4.3.3). These three models exploit different definitions of the event $\widehat{R}_{i,j} \in T$. In the *intensional* model (Section 4.3.1), the event $\widehat{R}_{i,j} \in T$ 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 (Section 4.3.2), the event $\widehat{R}_{i,j} \in T$ 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* model (Section4.3.3), is a combination of the other two models.

4.3.1 The *intensional* inductive model

In the *intensional inductive model*, we exploit direct probabilities to derive the induced probabilistic *intensional* definition $P_I(\widehat{R}_{i,j}|E)$. We evaluate this probability using the direct probability of $R_{i,j} \in T$ and the direct probabilities of having a transitive connection between i and j of two direct relations. For each possible node k, we then consider all alternative events $R_{i,k} \in T$ and $R_{k,j} \in T$. We use a running example to illustrate the idea.

We suppose to have four elements in a network (see Figure 4.3): "*lettuce*" (i), "*food*" (j), "*vegetable*" (k_1), and "*animal*" (k_2). We empirically estimated the *direct* probabilities (bold arrows) and we then determined the *induced* probability (dashed arrow). Both the $i - k_1 - j$ and $i - k_2 - j$ paths offer some information to the final induced probability even if we expect that $P(R_{i,k_1}|E)$, i.e., the direct probability of "*lettuce*" is a "*vegetable*", is closer to one and that $P(R_{i,k_2}|E)$, i.e., the direct probability of "*lettuce*" is a "*animal*", is closer to zero. We compute the induced probability as the probability of alternative events that represent the sub-part of the network of direct events. In this case, the induced probability is then:

$$P_I(\widehat{R}_{i,j}|E) = P(R_{i,j} \cup (R_{i,k_1} \cap R_{k_1,j}) \cup (R_{i,k_2} \cap R_{k_2,j})|E)$$

We can compute this probability using the inclusion-exclusion principle and some assumptions on the independence among events. The inclusion-exclusion principle gives the possibility of computing the probabilities of alternative events.

Given n probabilistic events $A_1, A_2, ..., A_n$ in a probability space, the probability of the union of these events is:

$$P(A_1 \cup A_2 \cup ... \cup A_n) = \sum_{\emptyset \neq J \subseteq \{1,...,n\}} (-1)^{|J|-1} P(A_J)$$

where $A_J = \bigcap_{i \in J} A_i$. The probability $P_I(\widehat{R}_{i,j}|E)$ can be then rewritten as:

$$\begin{aligned} P_I(\widehat{R}_{i,j}|E) = \quad & P(R_{i,j}|E)+P(R_{i,k_1}\cap R_{k_1,j}|E)+P(R_{i,k_2}\cap R_{k_2,j}|E)+ \\ & -P(R_{i,j}\cap R_{i,k_1}\cap R_{k_1,j}|E)-P(R_{i,k_1}\cap R_{k_1,j}\cap R_{i,k_2}\cap R_{k_2,j}|E)+ \\ & -P(R_{i,j}\cap R_{i,k_2}\cap R_{k_2,j}|E)+P(R_{i,j}\cap R_{i,k_1}\cap R_{k_1,j}\cap R_{i,k_2}\cap R_{k_2,j}|E) \end{aligned}$$

Finally, assuming that the probabilities of the direct events $R_{n,m}$ are independent, we can determine the probabilities of any of the joint events as the product of the probabilities of the events, e.g.: $P(R_{i,k_1}\cap R_{k_1,j}|E)=P(R_{i,k_1}|\vec{e}_{i,k_1})P(R_{k_1,j}|\vec{e}_{k_1,j})$.

The general equation for the induced intensional probability is the following:

$$P_I(\widehat{R}_{i,j}|E) = P(R_{i,j} \cup \bigcup_{k \in K} (R_{i,k} \cap R_{k,j})|E)$$

where $K = \{k_1, ..., k_n\}$ is the set of the intermediate nodes considered between i and j. As in the case of Equation 4.2, we can compute this equation using the inclusion-exclusion principle:

$$P_I(\widehat{R}_{i,j}|E) = \sum_{\emptyset \neq J \subseteq \{\epsilon, k_1, ..., k_n\}} (-1)^{|J|-1} P(R_J|E)$$

where $R_J = \bigcap_{k \in J} R_k$. Each R_k is defined as $R_\epsilon = R_{i,j}$ and $R_k = (R_{i,k} \cap R_{k,j})$ if $k \neq \epsilon$. Using the assumption that direct probabilities of $R_{m,n}$ are independent,

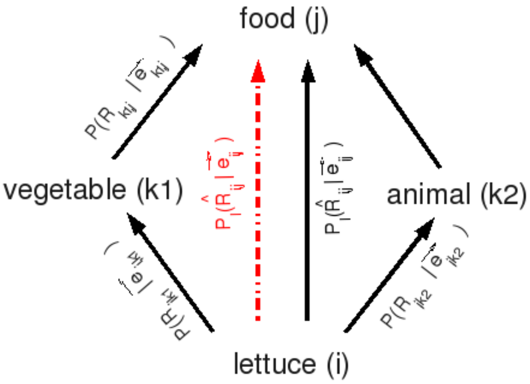


Figure 4.3: Example of *intensional* inductive model

we can also rewrite $P(R_J|E)$ as:

$$P(R_J|E) = \prod_{k \in J} P(R_k|E)$$

where $P(R_\epsilon|E) = P(R_{i,j}|\vec{e}_{i,j})$ and $P(R_k|E) = P(R_{i,k}|\vec{e}_{i,k})P(R_{k,j}|\vec{e}_{k,j})$ if $k \neq \epsilon$.

4.3.2 The *extensional* inductive model

The *extensional inductive model* exploits the *extensional* definitions of the concepts to derive the *induced probabilities*. Figure 4.4 reports an example where two different models are adopted. The first model (see Figure 4.4.(a)) uses the *extensional* definition of the two involved concepts, i.e., "*turkey*" and "*boat*", to determine the probability of the induced relation "*bird*" is a "*beast*", i.e., $P(\widehat{R}_{i,j}|E)$. The similarity between the extensional definition of "*bird*"(i) and of "*beast*"(j) should help in determining the probability of the relation between the

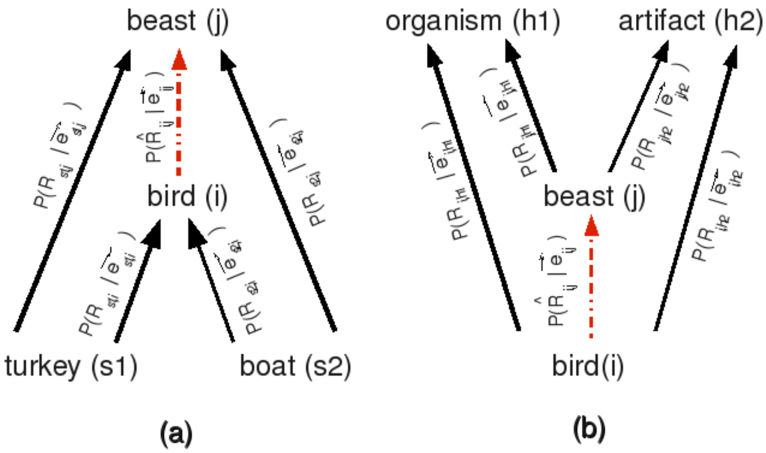

beast (j)
$P(R_{s1j} | \vec{e}_{s1j})$
$P(\hat{R}_{ij} | \vec{e}_{ij})$
$P(R_{s2j} | \vec{e}_{s2j})$
bird (i)
$P(R_{s1i} | \vec{e}_{s1i})$
$P(R_{s2i} | \vec{e}_{s2i})$
turkey (s1)
boat (s2)
(a)
organism (h1)
artifact (h2)
$P(R_{ih1} | \vec{e}_{ih1})$
$P(R_{jh1} | \vec{e}_{jh1})$
$P(R_{jh2} | \vec{e}_{jh2})$
$P(R_{ih2} | \vec{e}_{ih2})$
beast (j)
$P(\hat{R}_{ij} | \vec{e}_{ij})$
bird(i)
(b)

Figure 4.4: Example of *extensional induced* model

two concepts. In the second model (see Figure 4.4.(b)), "*animal*" and "*penguin*" contribute to the *extensional* definition of both *organism* and "*artifact*". This should help in determining the probability $P(\widehat{R}_{i,j}|E)$ of the induced event $\widehat{R}_{i,j}$. In the case of the reported running examples the probability of the induced event is:

$$P(\widehat{R}_{i,j}|E) = P(R_{i,j} \cup (R_{s_1,i} \cap R_{s_1,j}) \cup (R_{s_2,i} \cap R_{s_2,j}) \cup (R_{i,h_1} \cap R_{j,h_1}) \cup (R_{i,h_2} \cap R_{j,h_2})|E)$$

These probability equations can be reduced using the inclusion-exclusion principle and the independence assumption between the *direct* events. We can then rewrite this equation as:

$$\begin{aligned} P_E(\widehat{R}_{i,j}|E) = \ & P(R_{i,j}|E)+P(R_{s_1,i}\cap R_{s_1,j}|E)+P(R_{s_2,i}\cap R_{s_2,j}|E)+P(R_{i,h_1}\cap R_{j,h_1}|E)+ \\ & P(R_{i,h_2}\cap R_{j,h_2}|E)-P(R_{i,j}\cap R_{s_1,i}\cap R_{s_1,j}|E)-\cdots+ \\ & P(R_{i,j}\cap R_{s_1,i}\cap R_{s_1,j}\cap R_{s_2,i}\cap R_{s_2,j}|E)+\cdots+ \\ & -P(R_{i,j}\cap R_{s_1,i}\cap R_{s_1,j}\cap R_{s_2,i}\cap R_{s_2,j}\cap R_{i,h_1}\cap R_{j,h_1}|E)-\cdots+ \\ & +P(R_{i,j}\cap R_{s_1,i}\cap R_{s_1,j}\cap R_{s_2,i}\cap R_{s_2,j}\cap R_{i,h_1}\cap R_{j,h_1}\cap R_{i,h_2}\cap R_{j,h_2}|E) \end{aligned}$$

We can finally write the general equation using the *extensional* probabilistic definitions of the concepts. In this model we mix the two previous models in one single equation. The probability $P(\widehat{R}_{i,j}|E)$ of the induced event $\widehat{R}_{i,j}$ is then rewritten in term of the probabilities of the direct events as follows:

$$P_E(\widehat{R}_{i,j}|E) = P(R_{i,j} \cup \bigcup_s (R_{i,s} \cap R_{j,s}) \cup \bigcup_h (R_{h,i} \cap R_{h,j})|E)$$

4.3.3 The *mixed induced* model

The *mixed induced* model unifies the above mentioned methods, considering both the *intensional* and the *extensional* probabilistic models. Formally:

$$\begin{aligned} P_M(\widehat{R}_{i,k}|E) = P(R_{i,j} \cup \bigcup_k (R_{i,k} \cap R_{k,j}) \cup \\ \cup \bigcup_s (R_{i,s} \cap R_{j,s}) \cup \bigcup_h (R_{h,i} \cap R_{h,j})) \end{aligned}$$

Similarly, the inclusion-exclusion principle can be used to evaluate the alternative probability also for the *mixed* method.

The complete computation of the *induced* probabilistic models presented in this section is unfeasible as the computation of inclusion-exclusion principle is combinatorial with respect to the set of alternative events J. We then use an approximated computation derived from the method described in [Kahn *et al.*(1993)].

4.4 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 (Section 4.4.1). The second experiment is a full experiment that differs from the pilot in the size of semantic networks and in target relations (Section 4.4.2). For both sets of experiments we describe the experimental set up and we report the results.

4.4.1 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.

4.4.1.1 Corpus

As corpus we used the *English Web as Corpus* (ukWaC) [Baroni *et al.*(2009)] that we described in Section 3.5.1.1.

4.4.1.2 Semantic Networks

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[3] [Miller(1995)] as we described in Section 3.5.1.2.

4.4.1.3 Feature Space

We used a bag-of-n-gram feature space for implicitly modeling lexical-syntactic patterns, as defined in Section 3.5.2.3.

4.4.1.4 Logistic regressors

We used the logistic regressors defined in Chapter 3, i.e. a logistic regressor based on the Monroe-Penrose pseudo-inverse matrix [Golub and Kahan(1965)] as described in [Fallucchi and Zanzotto(2009a)].

4.4.1.5 Results

With the first set of experiments, we analyze the effectiveness of our *inductive* model. We evaluate the *iterative* (Section 2.4.2), the *direct* (Section 3.1), and the *induced* probabilistic models (Section 4.2) on their ability of sorting the

[3] We used the version 3.0 of WordNet

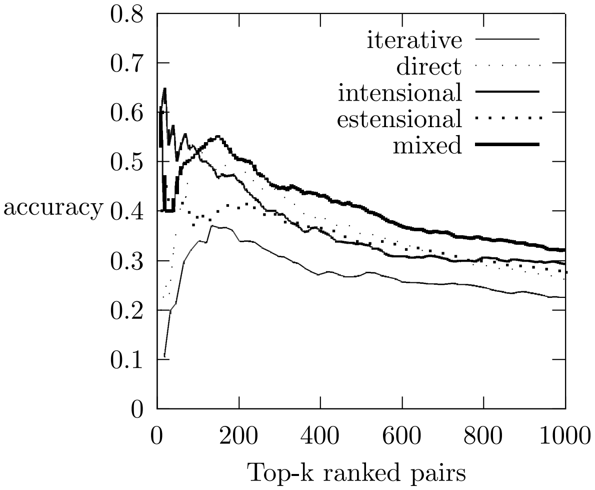


Figure 4.5: Accuracy of the top-k ranked pairs for the iterative, direct , and inductive probabilistic semantic networks learners

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.

The results are reported in Figure 4.5. Firstly, we can observe that, after

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

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.

The accuracies are reported in Table 4.1. 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.

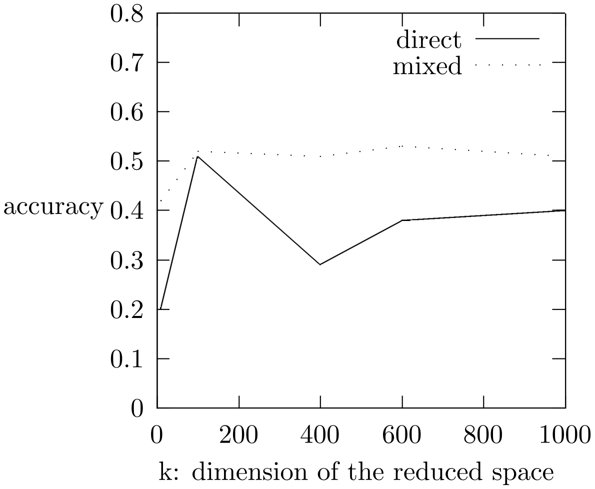


Figure 4.6: Accuracy of the direct and inductive probabilistic semantic networks learners with respect to SVD feature selection

With the second set of experiments, we want to investigate the role of the feature selection performed using SVD on our probabilistic model. Then, we analyze the accuracy on 100 considered pairs for different values of k, i.e., the number of considered dimensions for the SVD used in the computation of the pseudo-inverse matrix. The plots of the *direct* and the mixed inductive probabilistic models are presented in Figure 4.6. For both models, the performances are stable or decrease after k=100. An aggressive dimensionality reduction of

	eigenvector 1		*eigenvector 400*	
rank	feature	weight	feature	weight
1	,	$2.9363\ 10^{-4}$	clear	$86.1446\ 10^{-4}$
2	be	$0.5762\ 10^{-4}$	of ”	$54.8997\ 10^{-4}$
3	play	$0.2077\ 10^{-4}$	clear of	$47.1909\ 10^{-4}$
4	&	$0.1984\ 10^{-4}$	expedition	$40.7345\ 10^{-4}$
5	, as	$0.1965\ 10^{-4}$	burnt	$36.1784\ 10^{-4}$
6	-	$0.1671\ 10^{-4}$	),	$34.8534\ 10^{-4}$
7	is	$0.1356\ 10^{-4}$	tank	$32.9300\ 10^{-4}$
8	:	$0.0858\ 10^{-4}$	fishing	$31.8269\ 10^{-4}$
9	(	$0.0839\ 10^{-4}$	preparation	$31.4684\ 10^{-4}$
10	find	$0.0689\ 10^{-4}$	group	$31.2342\ 10^{-4}$

Table 4.2: Two selected eigenvectors on the bag-of-n-grams

the feature space does not negatively affect performances. For example, performances are not significantly affected if taking k=100 features instead of k=1000 features but the model are computed much faster. The stability of the two curves suggests that, even using the whole feature space, the performance cannot increase.

4.4.1.6 Qualitative analysis of dimensionality reduction

The experiments show that we can positively use dimensionality reduction of SVD within the computation of the pseudo-inverse matrix. We want now to analyze the first dimensions to understand which linear combination of the original features is relevant for the specific task of learning taxonomies using lexical patterns. As the decomposition algorithm we are using sorts the eigenvectors according to decreasing values of eigenvalues, we will examine the first eigenvector that should be more significant and the eigenvector number 400. In Table 4.2, we present only some of these eigenvectors. We present the dimensions with the 10 largest values. The first 10 dimensions of the first eigenvector are presented in column 2 and 3. The first 10 dimension of the 400th eigenvector are presented in column 4 and 5.

The first eigenvector is very interesting as it mixes many classical indicators of hypernymy, e.g., ",", "be", "&", etc. These indicators appear with different relative weights in many of the first eigenvectors. It is worth noticing that the forms of the verb *to be* are present in the considered eigenvector. On the contrary, the eigenvector number 400 does not contain any relevant information related to the hypernymy phenomenon in the first positions. This qualitatively explains what has been shown by the experiments in the previous section: many dimensions in the reduced space are totally irrelevant.

4.4.2 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 semantic network of words and a set of negative examples for the target relation; (2) a corpus for extracting evidences to derive probabilities; (3) the definition of the feature space; and, finally, (4) the definition of the logistic regressors.

4.4.2.1 Corpus

As corpus we used the *English Web as Corpus* (ukWaC) [Baroni *et al.*(2009)] that we described in Section 3.5.1.1.

4.4.2.2 Semantic Networks

The semantic network of words will be used as source of training and testing examples. For each experiment we need: a training example set $TR = (TR_p, TR_n)$ with positive pairs TR_p and negative pairs TR_n and a testing example set $TS = (TS_p, TS_n)$ with positive pairs TS_p and negative pairs TS_n. 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[4] [Miller(1995)]. In WordNet, semantic relations R are expressed as pairs of synonymy sets (synset), i.e., R=$\{(S_1, S_2)|S_1$ is in relation R with $S_2\}$ where the synset S_1 and S_2 are the sets of words $S_1 = \{w_1^{(1)}, \ldots, w_n^{(1)}\}$ and $S_2 = \{w_1^{(2)}, \ldots, w_m^{(2)}\}$. 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

[4] We use the version 3.0 in prolog.

Test	*Set*	*Description*	*Initial Size*	*Retreived Pairs*
isa	TR_p	$\mathcal{H}/\mathcal{H}_{ts}$	1983197	212076
	TR_n	$\overline{\mathcal{H}}/\overline{\mathcal{H}}_{ts}$	5594387	315428
	TS_p	$\mathcal{H}_{ts}$	506	150
	TS_n	$\overline{\mathcal{H}}_{ts}$	80436	258
part-of	TR_p	$\mathcal{M}/\mathcal{M}_{ts}$	14333	8077
	TR_n	$\overline{\mathcal{H}}/\mathcal{M}_{ts}$	623616	318679
	TS_p	$\mathcal{M}_{ts}$	408	101
	TS_n	$\overline{\mathcal{M}}_{ts}$	34214	1713

Table 4.3: Semantic networks used in the experiments

of words from the synset network.

Given one of the two target relations, we can derive the network of words $\mathcal{R}$ from the set R as follows: $\mathcal{R} = \{(w_a, w_b)|(S_a, S_b) \in R, w_a \in S_a, w_b \in S_b\}$. We then derived the semantic networks of words for hyperonymy $\mathcal{H}$ and for meronymy $\mathcal{M}$. These networks consist of, respectively, 7879350 and 672571 as reported in Table 4.3.

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

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*". We then obtained $\mathcal{H}_{ts}$, and $\mathcal{M}_{ts}$ respectively containing 506 and 408 pairs. Given the sets $W(veg)$ and $W(face)$ of the words respectively in $\mathcal{H}_{ts}$ and $\mathcal{M}_{ts}$, the negative examples are $\overline{\mathcal{H}}_{ts} = W(veg) \times W(veg) - \mathcal{H}_{ts}$, and $\overline{\mathcal{M}}_{ts} = W(face) \times W(face) - \mathcal{M}_{ts}$. In this way, we have the overall potential network of words for the testing.

The final sets are reported in Table 4.3. 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 lateron described.

4.4.2.3 Feature Space

We used a bag-of-n-gram feature space for implicitly modeling lexical-syntactic patterns such as defined in Section 3.5.2.3.

4.4.2.4 Logistic regressors

We used two different logistic regressors: a logistic regressor based on the Monroe-Penrose pseudo-inverse matrix (in Chapter 3) and the support vector machines [Vapnik(1995)] as implemented in [Joachims(1999)].

4.4.2.5 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

	direct		*intensional*		*extensional*		*mixed*	
	PI	SVM	PI	SVM	PI	SVM	PI	SVM
100	30.67	30.00	4.00	1.33	37.33	35.33	24.000	24.000
200	56.67	49.33	27.33	26.00	60.67	61.33	45.333	43.333
300	74.67	74.67	64.00	65.33	81.33	78.67	64.667	66.000

Table 4.4: Relative Recall of is-a relation: case semi-supervised

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 and Pennacchiotti(2006)]). The results of these experiments for the isa relation and the part-of relation are reported respectively in Table 4.4 and in Table 4.5. These tables report the *relative recall* of the different methods obtained using the first k ranked pairs. In line with [Pantel and 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
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 4.5: Relative Recall of part-of 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 4.4), we report the relative recall for the first 100, 200, and 300 first ranked pairs. For the part of relation, we report the relative recall for 500 and 1000 first ranked pairs. For the isa relation (Table 4.4), 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 4.4), 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

	direct	*intensional*	*extensional*	*mixed*
	PI	PI	PI	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 4.6: Relative Recall of is-a relation Vegetable: case supervised

supervised, assigning an initial probability of 0.99 to positive examples and an initial probability of 0.01 to negative examples.

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

Table 4.7: Relative Recall of part-of relation Face: case supervised

The results of these experiments for the isa and the part-of relations are reported respectively in Table 4.6 and in Table 4.7. 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 outperform the *intensional* model. For the part-of relation, both the *intensional* and the *extensional* models are suitable for exploiting transitivity.

4.5 Conclusion

We presented a probabilistic semantic networks learning model that positively exploits transitivity. We demonstrated that keeping the probability within the final knowledge base is extremely important for the performances of the learning method. We have also shown that our model positively exploits transitiveness as proved by the fact that the *inductive* model outperforms the *direct* model. Finally, we have demonstrated that SVD can be used as natural feature selection model within probabilistic semantic networks learning models.

5

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(1992b), Pantel and 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 not existing structured knowledge. The big challenge is to successful 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 is organized as follows. We present our model in Section 5.1. In section 5.2, we, then, evaluate and assess the performance of our method on the target domain, i.e., Earth Observation Domain. Finally, in section 5.3, we draw some conclusions.

5.1 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 [Snow *et al.*(2006), Fallucchi and Zanzotto(2009a)]. 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(1992a)]. 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 (Section 5.1.1) and we will then illustrate the method that we will be adapted to the new domain (Section 5.1.2).

5.1.1 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, $R_{dog,animal} \in O$ states that *dog* is an *animal* according to the ontology O.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \in O)$ of an assertion $R_{i,j}$ to belong to the ontology O and (2) the posterior probability $P(R_{i,j} \in O | \overrightarrow{e}_{i,j})$ 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 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(1992a)]). 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 $\widehat{O}$ that maximizes the probability of having the evidences of E, i.e.:

$$\widehat{O} = \arg\max_{O} P(E|O)$$

In the original model [Snow *et al.*(2006), Fallucchi and Zanzotto(2009a)], 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 *et al.*(2006), Fallucchi and Zanzotto(2009a)] this ratio is called *odds*. It is calculated using the logistic regression and then solving a linear problem using the pseudo-inverse matrix [Fallucchi and Zanzotto(2009a)]. The regression coefficients will be estimated as follows

$$\widehat{\beta} = X_{C_B}^{+} l \qquad (5.1)$$

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.

5.1.2 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 as in [Fallucchi and Zanzotto(2009a)]

$$l' = \alpha X_{C_A} \widehat{\beta} \tag{5.2}$$

where X_{C_A} is the inverse evidence matrix obtained from a *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.:

$$p_i = \frac{\exp(l_i)}{1 + \exp(l_i)} \tag{5.3}$$

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

5.2 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.

5.2.1 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[1] [Miller(1995)] and we obtained about 2 millions of words.

Here, we firstly define the semantic networks used in the experiments of Section 5.2.3. 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 TR_p and negative pairs TR_n, 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.

[1] We used the version 3.0 of WordNet

We extract the training example pairs from an existing knowledge repository: WordNet[2] [Miller(1995)]. Given hyperonymy as target relation, we can derive the network of words $\mathcal{R}$ from the set R as follows: $\mathcal{R} = \{(w_a, w_b)|(S_a, S_b) \in R, w_a \in S_a, w_b \in S_b\}$. We then build the set $\mathcal{H}$ that contains all pairs of words in WordNet that are in hyperonymy relation. Then $TR_p = \mathcal{H} - \mathcal{TS}$. 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 *et al.*(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 and 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 $T \cup \overline{T}$, 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 C_A 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.

[2] We use the version 3.0 in prolog.

5.2.2 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** A_o 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 and Poesio(2008)] we define the agreement value agr_i for all items I as follows:

$$arg_i = \begin{cases} 1 & \text{if annotators assign i to the same category} \\ 0 & \text{if annotators assign i to different categories} \end{cases}$$

The observed agreement has been evaluated as in the following:

$$A_o = \frac{1}{i} \sum_{i \in I} agr_i$$

This measure does not take into account changes in the agreement between two annotators. An improved measure of inter-annotator agreement is given by the Cohen's **kappa** coefficient [Cohen(1960)]. It is a statistical measure that takes into account the effect of changes in the agreement giving the possible agreement beyond change actually observed. The kappa-coefficient is defined as follows:

$$k = \frac{A_o - A_e}{1 - A_e}$$

- A_e : expected agreement by change
- $1 - A_e$: attainabled agreement over and above change
- $A_o - A_e$: actually found agreement beyond change

The expected agreement (A_e) is the probability of the agreement among annotators due to change. There are two different methods for estimating a probability distribution for random assignment of categories. The two approaches reflect different conceptualizations of the problem.

In the first approach, each annotator has a personal distribution, based on that annotator's distribution of categories [Cohen(1960)]. In the second approach, there is one distribution for all annotators, derived from the total proportions of categories assigned by all annotators [Scott(1955), Fleiss *et al.*(1971)]. Data are respectively visualized in a *contingency table* (first approach) and in an *agreement table* (second approach).

The distinction between the two approaches, in the case of two annotators, is often glossed over because in practice the two computations of A_e produce very similar (when not the same) outcomes, as shown in section 5.2.2.1. In

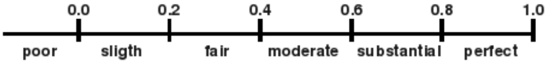


Figure 5.1: Scale for the interpretation of Kappa by Landis and Koch (1977)

[Carletta(1996)] the adaptation of the kappa coefficient to computational linguistic is suggested. Different levels of agreement may be defined, according to the experiments of a specific application. In [Landis and Koch(1977)] confidence intervals are proposed for the values of the kappa coefficient, as reported in Figure 5.1.

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 : **pairwise agreement** and **multi-π agreement** both in [Fleiss *et al.*(1971)]. In the first section 5.2.2.1 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-π agreement, we examine the distribution of all the three annotators.

5.2.2.1 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.

Given the same data (641 or 404-annotations) with the same guidelines, we build the contingency tables for the 3 pairwise annotators(respectly Table 2 and Table 4). For each table we report the statistic of the two annotators. Then in Table 1a 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 judgement and dividing by the total number of annotations. For example, considering $pair_1$ (first row of the Table 1a), 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

		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_3	yes	72	87	159
	no	36	446	482
		180	533	641

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

Table 1: Contingency tables for pairwise annotator agreement for 641-annotations

	A_o	A_e	$kappa$
$pair_1 = (A_1, A_2)$	0.8377535	0.7384206	0.3797428
$pair_2 = (A_1, A_3)$	0.8486739	0.6811997	0.5253266
$pair_3 = (A_2, A_3)$	0.8081123	0.6670496	0.4236749

Table 2: pairwise agreement for 641-annotations

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 $pair_1$ is $kappa = (0.8377535 - 0.7384206)/(1 - 0.7384206) = 0.3797428$. 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 3a). Summarizing only for $pair_3$ on 641-annotations the coefficient kappa is in the "fair" interval in accord to the scale proposed in [Landis and Koch(1977)] and reported in Figure 5.1. 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.

5.2.2.2 Multi-π agreement

In multi-π agreement the agreement of the annotators is considered as a whole. There is only one distribution for all the annotators, derived from the total

		A_1		
		yes	no	
A_2	yes	40	32	72
	no	35	297	332
		75	329	404

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

		A_1		
		yes	no	
A_3	yes	65	54	119
	no	10	275	285
		75	329	404

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

		A_2		
		yes	no	
A_3	yes	53	66	119
	no	19	266	285
		72	332	404

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

Table 3: Contingency tables: pairwise annotator agreement for 404-annotations

	A_o	A_e	$kappa$
$pair_1 = (A_1, A_2)$	0.8341584	0.7023086	0.4429077
$pair_2 = (A_1, A_3)$	0.8415842	0.6291663	0.5728117
$pair_3 = (A_2, A_3)$	0.7896040	0.6322174	0.4279336

Table 4: pairwise agreement for 404-annotations

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 4a and table 4b report 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 4c 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.

5.2.3 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 5a and in Table 5b.

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

	A_o	A_e	*kappa*
641-annotations	0.83151	0.69764	0.44277
404-annotations	0.82382	0.65739	0.48577

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

Table 5: Agreement tables and Multi-π agreement for 641 and 404 annotations

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 6: Performance of both systems with respect to 3 annotators

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.*

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.

5.3 Conclusion

In this Chapter we present an ontology learning method that can exploit the models learned from a generic domain to extract new information in a specific domain. In our model, we firstly learn a model from the training data, then we use the learned model to discover the relation between two words in a specific domain.

We tested our model adaptation strategy using a *background* domain that is applied to learn the *isa* networks in a specific domain, i.e., the Earth Observation Domain. The results of the experiments show that this way of using a model identified in a *background* domain is helpful to learn the *isa* relation in Earth Observation Domain.

6

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 *et al.*(2006)]). Studying methods and building systems for 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 *et al.*(2005), Navigli and Velardi(2004)] as well as knowledge harvesting methods in NLP either [Hearst(1992a), Pantel and Pennacchiotti(2006)]. These learning methods use variants of the distributional hypothesis [Harris(1964)] or exploit some induced lexical-syntactic patterns [Robison(1970)]. The task is generally seen as a classification (e.g., [Pekar and Staab(2002), Snow *et al.*(2006)]) or a clustering (e.g., [Cimiano *et al.*(2005)]) problem. This allows the use of both machine learning and probabilistic models.

Models for automatic creating knowledge repositories generally exploit existing structured knowledge such as existing thesauri. Methods based on the Hearst's work [Hearst(1992a)] use existing pairs of words in a given semantic relation to extract patterns from corpora. These patterns are then used to induce novel pairs of words that are in the same semantic relation. For example, the pair of words *Bush* and *New Haven* are known examples of the semantic relation *has_born_in*. These part can be used to deduce from corpora that *is the birthplace of* is a good pattern to induce other instances of the above relation. Yet, these models cannot be easily used to exploit the formal properties of the target relation and, for this reason, they cannot properly exploit information derived indirectly for existing data.

Some semantic relations such as hyperonymy and part-of have an extremely important property that is transitiveness. This property, along with the use of existing knowledge repositories, may help to build better knowledge extraction and structuring models during the discovering phase. Such idea is explored in [Snow *et al.*(2006), Fallucchi and Zanzotto(2009a)].

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 and Volker(2005)]. In the following, we present the Semantic Turkey Ontology Learner (ST-OL) [Fallucchi *et al.*(2009)], an incremental ontology learning system that follows the above idea putting final users in the learning loop. Furthermore, this system uses a 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 (Section 6.1). We then introduce ST-OL, the system that we have adopted following the above principles (Section 6.2). Finally, we draw some conclusions (Section 5.3).

6.1 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 (Section 6.1.1). Then, we describe the used ontology editor (Section 6.1.2). Finally, we introduce the adopted ontology learning methodology (Section 6.1.3).

6.1.1 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 ontologies $O_0 \dots O_n$. 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:

$$M_C(O_i, \overline{O}_i) = \widehat{O}_{i+1} \overset{UV}{\rightsquigarrow} (O_{i+1}, \overline{O}_{i+1}) \tag{6.1}$$

where M_C is the model learnt from the corpus, O_i is the ontology at the $i - th$ step and $\overline{O}_i$ 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 $\widehat{O}_{i+1}$. The UV on the first k possibilities produces the updated ontology O_{i+1} and the updated *non-ontology* $\overline{O}_{i+1}$. At the initial step, the process has O_0 and $\overline{O}_0 = \emptyset$. The *ontology learner* produces the model $M_C(O_i, \overline{O}_i)$ 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* $\overline{O}_i$.

6.1.2 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 *et al.*(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 partitive 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 *et al.*(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.

6.1.3 Probabilistic Ontology Learner

We use the Probabilistic Ontology Learning (POL) [Fallucchi and Zanzotto(2009a)] 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 *et al.*(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 $R_{i,j}$. 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 (i.e., the direct or the indirect generalization). For example, $R_{dog,animal} \in O$ describes that *dog* is an *animal* according to the ontology O.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \in O)$ of an assertion $R_{i,j}$ to belong to the ontology O and (2) the posterior probability

$P(R_{i,j} \in O|\overrightarrow{e}_{i,j})$ 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 j (see [Hearst(1992a)]).

Given a set of evidences E over all the relevant word pairs, in [Snow *et al.*(2006), Fallucchi and Zanzotto(2009a)] the probabilistic ontology learning task is defined as the problem of finding an ontology $\widehat{O}$ that maximizes the probability of having the evidences E, i.e.:

$$\widehat{O} = \arg\max_{O} P(E|O)$$

In the original model [Snow *et al.*(2006), Fallucchi and Zanzotto(2009a)], 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 $\Delta(N)$ and is defined as follows:

$$\Delta(N) = P(E|O')/P(E|O) \tag{6.2}$$

It is also possible to demonstrate that

$$\begin{aligned}\Delta(R_{i,j}) &= k \cdot \frac{P(R_{i,j} \in O|\overrightarrow{e}_{i,j})}{1 - P(R_{i,j} \in O|\overrightarrow{e}_{i,j})} = \\ &= k \cdot odds(R_{i,j})\end{aligned}$$

where k is a constant (see [Snow *et al.*(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 (Chapter 3)

$$\widehat{\beta} = X^{+}l \tag{6.3}$$

where $\widehat{\beta}$ 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 (3.7) and the equation (6.3):

$$XX^{+}\, l_i = \widehat{l}_{i+1} \overset{UV}{\rightsquigarrow} l_{i+1} \tag{6.4}$$

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 (6.1). The matrix XX^+ 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 $\overline{O_i}$ as it includes the logit of both probabilities (0.99 and 0.01).

6.2 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 workflow 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* (c.f., Section 6.1.1), 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 6.1). 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 6.1 shows an example of an enriched initial ontology.

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.

In order to focus, if possible, only on good pairs, ST-OL shows only pairs above a threshold τ of probabilities. For example, in Figure 6.1, 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

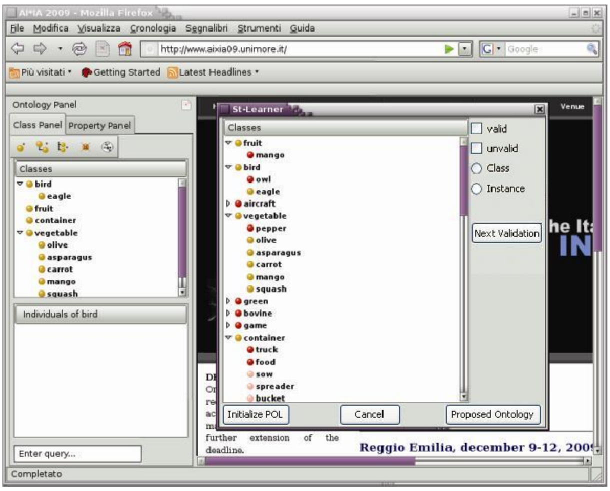


Figure 6.1: Initial Ontology extended with the pairs proposed by the POL System

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 6.2 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 presented in Section 6.1.3, this

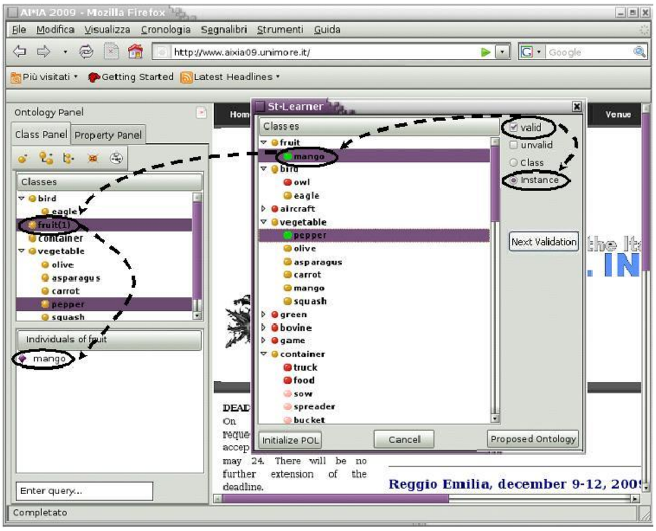


Figure 6.2: Manual validation of new resources added to the ontology

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.

6.3 Conclusion

In this Chapter, we presented a computational model POL and a system ST-OL for incremental ontology learning. POL is basically an incremental probabilistic model to learn ontological information from texts and it is designed to positively exploit a probabilistic ontology learning method within a learning loop that includes final users. ST-OL, being developed and integrated as an extension for the Knowledge Management and Acquisition platform Semantic Turkey, has inherited all the facilities that the main application is providing for ontology development, as well as those exposed by the hosting Web Browser (which enabled, for example, to rapidly integrate a web spider into the application and use it to provide corpora for learning probabilistic models and/or for inducing new ontology contributions). ST-OL (and Semantic Turkey as its founding technology) has thus proved to be the right environment for embodying this kind of process, providing the crossroads between Users, Web and Knowledge.

7

Conclusions and Future Works

Describing word meaning is one of the most interesting challenges of natural language processing as texts can not 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 thesis 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 SVD-based logistic approach has proved to be efficient and our probabilistic model suitable for exploiting the structural properties of semantic relations in learning semantic networks. As a side effect, we also demonstrated that SVD can be used as natural feature selection model within probabilistic taxonomy learning models.

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.

Bibliography

[Ando(2004)] Ando, R. K. (2004). Exploiting unannotated corpora for tagging and chunking. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Morristown, NJ, USA. Association for Computational Linguistics.

[Artstein and Poesio(2008)] Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Journal of Computational Linguistics*, **34**(4).

[Bacchiani *et al.*(2004)] Bacchiani, M., Roark, B., and Saraclar, M. (2004). Language model adaptation with map estimation and the perceptron algorithm. In D. M. Susan Dumais and S. Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 21–24, Boston, Massachusetts, USA. Association for Computational Linguistics.

[Bagni *et al.*(2007)] Bagni, D., Cappella, M., Pazienza, M. T., Pennacchiotti, M., and Stellato, A. (2007). Harvesting relational and structured knowledge for ontology building in the wpro architecture. In R. Basili and M. T. Pazienza, editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing, 10th Congress of the Italian Association for Artificial Intelligence, Rome, Italy, September 10-13, 2007, Proceedings. Lecture Notes in Computer Science*, volume 4733, pages 157–169. Springer.

[Baroni and Bisi(2004)] Baroni, M. and Bisi, S. (2004). Using cooccurrence statistics and the web to discover synonyms in technical language. In *In*

Proceedings of LREC 2004, pages 1725–1728.

[Baroni *et al.*(2009)] Baroni, M., Bernardini, 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, Parts 3**, 209–226.

[Basili *et al.*(2007)] Basili, R., Gliozzo, A., and Pennacchiotti, M. (2007). Harvesting ontologies from open domain corpora: a dynamic approach. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*, Borovets, Bulgaria.

[Berners-Lee *et al.*(2001)] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, **284**(5), 34–43.

[Bertoldi and Federico(2009)] Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 182–189, Morristown, NJ, USA. Association for Computational Linguistics.

[Blitzer *et al.*(2006)] Blitzer, J., Mcdonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing.*

[Bourigault *et al.*(2001)] Bourigault, D., Jacquemin, C., and L, M.-C. (2001). *Recent Advances in Computational Terminology.* John Benjamins, Amsterdam.

[Buitelaar and Sacaleanu(2002)] Buitelaar, P. and Sacaleanu, B. (2002). Extending synsets with medical terms. In *Proceedings of the first international conference on global WordNet, Mysore, India*, pages 21–25.

[Buitelaar *et al.*(2004)] Buitelaar, P., Olejnik, D., and Sintek, M. (2004). A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In *1st European Semantic Web Symposium (ESWS)*, Heraklion, Greece.

[Buitelaar *et al.*(2005)] Buitelaar, P., Cimiano, P., and Magnini, B., editors (2005). *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

[Burger and Ferro(2005)] Burger, J. and Ferro, L. (2005). Generating an entailment corpus from news headlines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54, Ann Arbor, Michigan. Association for Computational Linguistics.

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

[Caron *et al.*(1988)] Caron, D., Hospital, W., and Corey, P. N. (1988). Variance estimation of linear regression coefficients in complex sampling situation. *Sampling Error: Methodology, Software and Application*, pages 688–694.

[Chan and Ng(2007)] Chan, Y. S. and Ng, H. T. (2007). Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.

[Chelba and Acero(2006)] Chelba, C. and Acero, A. (2006). Adaptation of maximum entropy capitalizer: Little data can help a lot.

[Chklovski and Pantel(2004)] Chklovski, T. and Pantel, P. (2004). VerbOCEAN: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcellona, Spain.

[Cimiano and Volker(2005)] Cimiano, P. and Volker, J. (2005). Text2onto - a framework for ontology learning and data-driven change discovery. In A. Montoyo, R. Munoz, and E. Metais, editors, *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain. Springer.

[Cimiano *et al.*(2004)] Cimiano, P., Hotho, A., and Staab, S. (2004). Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text.

[Cimiano *et al.*(2005)] Cimiano, P., Hotho, A., and Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence research*, **24**, 305–339.

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

[Cortes and Vapnik(1995)] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 1–25.

[Cox(1958)] Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, **20**(2), 215–242.

[Dagan and Glickman(2005)] Dagan, I. and Glickman, O. (2005). The pascal recognising textual entailment challenge. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

[Daumé and Marcu(2006)] Daumé, III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, **26**, 101–126.

[Deerwester *et al.*(1990)] Deerwester, S., Dumais, S. T., Furnas, G. W., L, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**, 391–407.

[Dhillon *et al.*(2003)] Dhillon, I. S., Mallela, S., Guyon, I., and Elisseeff, A. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, **3**, 2003.

[Dolan *et al.*(2004)] Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Coling 2004*, pages 350–356, Geneva, Switzerland. COLING.

[Etzioni *et al.*(2004)] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. (2004). Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of* *the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA. ACM.

[Evans(2003)] Evans, R. (2003). A framework for named entity recognition in the open domain. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2003)*, pages 137 – 144, Borovetz, Bulgaria.

[Fallucchi and Zanzotto(2009a)] Fallucchi, F. and Zanzotto, F. M. (2009a). SVD feature selection for probabilistic taxonomy learning. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 66–73, Athens, Greece. Association for Computational Linguistics.

[Fallucchi and Zanzotto(2009b)] Fallucchi, F. and Zanzotto, F. M. (2009b). Svd for feature selection in taxonomy learning. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*, Borovets, Bulgaria. John Benjamins.

[Fallucchi and Zanzotto(2010)] Fallucchi, F. and Zanzotto, F. M. (in print 2010). Inductive probabilistic taxonomy learning using singular value decomposition. *In Journal of Natural Language Engineering.*

[Fallucchi *et al.*(2009)] Fallucchi, F., Scarpato, N., Stellato, A., and Zanzotto, F. M. (2009). Probabilistic ontology learner in semantic turkey. In *AI*IA '09:: Proceedings of the XIth International Conference of the Italian Association for Artificial Intelligence Reggio Emilia on Emergent Perspectives in Artificial Intelligence*, pages 294–303, Berlin, Heidelberg. Springer-Verlag.

[Fleiss *et al.*(1971)] Fleiss, J. *et al.* (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, **76**(5), 378–382.

[Frantzi *et al.*(2000)] Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, **3**(2), 115–130.

[Gao(2009)] Gao, J., W. Q. B. C. S. K. S. Y. K. N. e. a. (2009). Model adaptation via model interpolation and boosting for web search ranking. In *Conference on Empirical Methods in Natural Language Processing*.

[Gauvain and Lee(1994)] Gauvain, J.-l. and Lee, C.-h. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, **2**, 291–298.

[Geffet and Dagan(2005)] Geffet, M. and Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 107–114, Morristown, NJ, USA. Association for Computational Linguistics.

[Gennari *et al.*(2003)] Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubzy, M., and 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(2001)] Gildea, D. (2001). Corpus variation and parser performance.

[Golub and Kahan(1965)] Golub, G. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for*

Industrial and Applied Mathematics, Series B: Numerical Analysis, **2**(2), 205–224.

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

[Griesi *et al.*(2006)] Griesi, D., Pazienza, M. T., and Stellato, A. (2006). Gobbleing over the web with semantic turkey. In *Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop (SWAP2006)*, Scuola Normale Superiore, Pisa, Italy,. 18-20 December.

[Griesi *et al.*(2007)] Griesi, D., Pazienza, M. T., and Stellato, A. (2007). Semantic turkey - a semantic bookmarking tool (system description). In E. Franconi, M. Kifer, and W. May, editors, *4th European Semantic Web Conference (ESWC 2007)*, volume The Semantic Web: Research and Applications, 4519 of *Lecture Notes in Computer Science*, pages 779–788, Innsbruck, Austria. Springer. Innsbruck, Austria, June 3-7.

[Gruber(1993)] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**(2), 199–220.

[Guyon and Elisseeff(2003)] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157–1182.

[Haase *et al.*(2008)] Haase, P., Lewen, H., Studer, R., Tran, D. T., Erdmann, M., d'Aquin, M., and Motta, E. (April, 2008). The neon ontology engineering toolkit. In *In WWW 2008 Developers Track.*

[Harris(1964)] Harris, Z. (1964). Distributional structure. In J. J. Katz and J. A. Fodor, editors, *The Philosophy of Linguistics*, New York. Oxford University Press.

[Harris(1968)] Harris, Z. (1968). *Mathematical structures of language.* Wiley.

[Hearst(1992a)] Hearst, M. A. (1992a). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics (CoLing-92)*, Nantes, France.

[Hearst(1992b)] Hearst, M. A. (1992b). Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 15th CoLing*, Nantes, France.

[Hindle(1990)] Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proc. of the 28th Annual Meeting on Association for Computational Linguistics*, pages 268–275.

[Jacquemin(2001)] Jacquemin, C. (2001). *Spotting and Discovering Terms through Natural Language Processing.* Massachusetts Institue of Technology, Cambrige, Massachussetts, USA.

[Joachims(1999)] Joachims, T. (1999). Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning.* MIT Press.

[Kahn *et al.*(1993)] Kahn, J., Linial, N., and Samorodnitsky, A. (1993). Inclusion-exclusion: Exact and approximate. *Combinatorica*, **16**, 465–477.

[Landauer and Dumais(1997)] Landauer, T. K. and Dumais, S. T. (1997). Solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, (104).

[Landis and Koch(1977)] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, **33**(1), 159–174.

[Lapata and Keller(2004)] Lapata, M. and 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 the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, MA.

[Lin and Pantel(2001a)] Lin, D. and Pantel, P. (2001a). DIRT-discovery of inference rules from text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*. San Francisco, CA.

[Lin and Pantel(2001b)] Lin, D. and Pantel, P. (2001b). Induction of semantic classes from natural language text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–322, New York, NY, USA. ACM.

[Lin and Pantel(2002)] Lin, D. and Pantel, P. (2002). Concept discovery from text.

[Liu(2007)] Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer.

[Maedche and Staab(2000)] Maedche, A. and Staab, S. (2000). Discovering conceptual relations from text. In *ECAI-2000 - European Conference on Artificial Intelligence*. IOS Press, Amsterdam.

[Maedche and Staab(2001)] Maedche, A. and Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems*, **16**(2), 72–79.

[Maedche and Staab(2002)] Maedche, A. and Staab, S. (2002). Measuring similarity between ontologies. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK. Springer-Verlag.

[Maedche and Volz(2001)] Maedche, A. and Volz, R. (2001). Icdm workshop on integrating data mining and knowledge management. In *The Text-To-Onto Ontology Extraction and Maintenance Environment*, San Jose, California, USA.

[McCarthy and Navigli(2007)] McCarthy, D. and Navigli, R. (2007). Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the* 4^{th} *International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic.

[McCarthy *et al.*(2004)] McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding predominant word senses in untagged text. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational*

Linguistics, page 279, Morristown, NJ, USA. Association for Computational Linguistics.

[Medche(2002)] Medche, A. (2002). *Ontology Learning for the Semantic Web*, volume 665 of *Engineering and Computer Science*. Kluwer International.

[Miller(1995)] Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, **38**(11), 39–41.

[Morin(1999)] Morin, E. (1999). *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. Ph.D. thesis, Univesité de Nantes, Faculté des Sciences et de Techniques.

[Navigli and Velardi(2004)] Navigli, R. and Velardi, P. (2004). Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.*, **30**(2), 151–179.

[Neches *et al.*(1991)] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Mag.*, **12**(3), 36–56.

[Nelder and Wedderburn(1972)] Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, **135**(3), 370–384.

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

[Pantel and Lin(2001)] Pantel, P. and Lin, D. (2001). A statistical corpus-based term extractor. In *AI '01: Proceedings of the 14th Biennial Conference of* *the Canadian Society on Computational Studies of Intelligence*, pages 36–46, London, UK. Springer-Verlag.

[Pantel and Pennacchiotti(2006)] Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia. Association for Computational Linguistics.

[Pedersen *et al.*(2004)] Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet::similarity - measuring the relatedness of concepts. In *Proc. of 5th NAACL*, Boston, MA.

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

[Penrose(1955)] Penrose, R. (1955). A generalized inverse for matrices. In *Proc. Cambridge Philosophical Society*.

[Ravichandran and Hovy(2002)] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania.

[Resnik(1993)] Resnik, P. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

[Roark and Bacchiani(2003)] Roark, B. and Bacchiani, M. (2003). Supervised and unsupervised pcfg adaptation to novel domains. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 126–133, Morristown, NJ, USA. Association for Computational Linguistics.

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

[Salton and Buckley(1987)] Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

[Scott(1955)] Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly*, **19**(3), 321–325.

[Snow *et al.*(2006)] Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *In ACL*, pages 801–808.

[Szpektor *et al.*(2004)] Szpektor, I., Tanev, H., Dagan, I., and Coppola, B. (2004). Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcellona, Spain.

[Toumouth *et al.*(2006)] Toumouth, A., Lehireche, A., Widdows, D., and Malki, M. (2006). Adapting wordnet to the medical domain using lexicosyntactic patterns in the ohsumed corpus. In *AICCSA '06: Proceedings of the IEEE International Conference on Computer Systems and Applications*, pages 1029–1036, Washington, DC, USA. IEEE Computer Society.

[Turcato *et al.*(2000)] Turcato, D., Popowich, F., Toole, J., Fass, D., Nicholson, D., and Tisher, G. (2000). Adapting a synonym database to specific domains. In *Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval*, pages 1–11, Morristown, NJ, USA. Association for Computational Linguistics.

[Turney(2001)] Turney, P. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl.

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

[Yeh(2000)] Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics*, pages 947–953, Morristown, NJ, USA. Association for Computational Linguistics.

[Yoshida *et al.*(2007)] Yoshida, K., Tsuruoka, Y., Miyao, Y., and ichi Tsujii, J. (2007). Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In M. M. Veloso, editor, *IJCAI*, pages 1783–1788.

[Zanzotto *et al.*(2006)] Zanzotto, F. M., Pennacchiotti, M., and Pazienza, M. T. (2006). Discovering asymmetric entailment relations between verbs using selectional preferences. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 849–856, MORRISTOWN, NJ – USA. Association for Computational Linguistics.