Abstract: We present a hybrid method for automated on-demand creation of conceptual models of domain-specific knowledge. Models are thereby created using a two-step process of Domain Definition and Domain Description. Domain Definition creates a conceptual base whereas in the Domain Description relationships are added to the conceptual model using a pattern-based relational-targeting Information Extraction algorithm. The two-step process has the advantage over traditional approaches to ontology learning that it provides conceptual grounding through a top-down extraction and over information extraction that the extraction operates on a conceptual level so that concept integrity and reference are guaranteed. At the core of the extraction algorithm is a novel measure for semantic overlap of relationships that allows the extraction of multiple intensionally similar relationships while disambiguating merely extensionally similar relationships. The envisioned use of the created models is primarily in Information Retrieval applications, but the models can also serve as starting points for formal ontologies in Knowledge Representation applications. We detail the techniques involved in domain definition and description as well as evaluate the outcomes in depth using qualitative and quantitative evaluation metrics.
November 30, 2102

Dear JWS EICs and Area Editors:

Please find our submission

On Demand Creation of Focused Domain Models using Topdown and Bottomup Information Extraction,
co-authored by Christopher Thomas, Pankaj Mehra, Amit Sheth, and Wenbo Wang.

Chris Bizer is probably the area editor closest to this work, but of course please consider assigning any area editor you deem to be closest to this work.

Our submission meets the standards of originality and is not under consideration anywhere else.

Please contact me if any additional information is required related to the submission.

Sincerely yours,

Prof. Amit Sheth  amit@knoesis.org  http://knoesis.org
LexisNexis Ohio Eminent Scholar
Director, Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis)
377 Joshi Research Center, Wright State University
3640 Colonel Glenn Hwy., Dayton, Ohio 45435-0001
On-Demand Creation of Focused Domain Models using Top-down and Bottom-up Information Extraction

Christopher Thomas  
Kno.e.sis Center  
Wright State University  
Dayton, OH  
topher@knoesis.org

Pankaj Mehra  
Whodini, Inc  
Los Altos, CA  
pankaj.mehra@ieee.org

Amit Sheth  
Kno.e.sis Center  
Wright State University  
Dayton, OH  
amit@knoesis.org

Wenbo Wang  
Kno.e.sis Center  
Wright State University  
Dayton, OH  
wenbo@knoesis.org

ABSTRACT

We present a hybrid method for automated on-demand creation of conceptual models of domain-specific knowledge. Models are thereby created using a two-step process of Domain Definition and Domain Description. Domain Definition creates a conceptual base whereas in the Domain Description relationships are added to the conceptual model using a pattern-based relational-targeting Information Extraction algorithm that deploys a novel relational pertinence measure to disambiguate semantically overlapping types of relationships. The two-step process has the advantage over traditional approaches to ontology learning that it provides conceptual grounding through a top-down extraction and over information extraction that the extraction operates on a conceptual level so that concept integrity and reference are guaranteed. At the core of the extraction algorithm is a novel measure for semantic overlap of relationships that allows the extraction of multiple intensionally similar relationships while disambiguating merely extensionally similar relationships. The envisioned use of the created models is primarily in Information Retrieval applications, but the models can also serve as starting points for formal ontologies in Knowledge Representation applications. We detail the techniques involved in domain definition and description as well as evaluate the outcomes in depth using qualitative and quantitative evaluation metrics.

1. INTRODUCTION

Conceptual search, browsing and classification of documents using background knowledge from domain models have been the topic of extensive research in the Semantic Web and Information Retrieval (IR) communities [11, 29]. Large search engines, such as Google and Bing have recently started to display factual knowledge that immediately pertains to the search query alongside the results. However, a limited number of concepts that immediately match the search terms is too narrow for a proper conceptual definition of a domain. Contextual browsing or document filtering needs models that are concise, but still comprehensive. For example, a Google or Bing search for “India” brings up, alongside the relevant web results, a short textual description of the country of India with some factual data (e.g. population, capital city), taken from Wikipedia. Once the search is expanded, though, e.g. to “India Pakistan Kashmir”, only web search results are displayed. Given the same query, the Doozer++ system described in this work produces a concise domain model that encompasses many concepts pertinent to the topic (See Figure 1). It identifies that Kashmir is a geographical region that may belong to both countries and even separates it from the geopolitical regions of “Azad Kashmir” and “Jammu and Kashmir” that politically belong to Pakistan and India, respectively. It also identifies pertinent heads of state, past and present. Such a model can be used to either explore the domain by reading the concept descriptions in the model or by using it to further filter Web search results according to their relevance to the model rather than just to the query.

A broader adoption of conceptual search for general purpose domains is largely dependent on the availability of customized and focused semantic domain models or ontologies. However, focused ontologies, domain models, controlled domain vocabularies and taxonomies that can be used for annotation and retrieval tasks are often either not available at all or expensive to obtain. The Taxonomy warehouse portal\(^1\),

\(^1\)http://www.taxonomywarehouse.com
Figure 1: Excerpt of the model about the relationship between India and Pakistan

for example, offers a variety of taxonomies for purchase. Apart from the associated cost that can often exceed tens of thousands of dollars per taxonomy, these vocabularies are still static and topic-centered instead of being dynamically developed for a specific user’s interests at a particular point in time. Large general purpose knowledge repositories, such as DBpedia [4] or Freebase [5], on the other hand, are too broad to be useful as browsing or classification aides. An automated solution that can deliver customized models on-demand in an affordable and scalable manner is thus highly desirable.

Current approaches to on-demand extraction of domain knowledge can broadly be classified along the two dimensions of extraction goals and information sources. The extraction goals encompass the envisioned application for the extracted information or knowledge as well as the formal requirements for the resulting model. Ontology Learning efforts usually attempt to create formally valid models whereas Information Extraction (IE) approaches place more emphasis on the throughput, precision and recall of the algorithm. Information sources can range between free text and formal concept descriptions. Free text itself can take the form of colloquial, abbreviated and ungrammatical language, such as microblog posts or it can be well-formed as is the case in peer-reviewed journals or in Wikipedia articles. Both the formal and informal information sources come with their own caveats: A top-down extraction from existing conceptual models suffers from the same sparsity of relational information that the original conceptual model had. Bottom-up extraction, on the other hand, tends to suffer from poor concept integrity. We define concept integrity as a consistent and grounded representation of classes, entities and relationships in such a way that an identifier that is used for a class, a relationship or an individual always refers to the same concept. The term concept is used here in a general definition as a unit of knowledge that comprises classes, individuals or relationship/property types in a domain model.

With the domain model creation application Doozer++, this paper presents a hybrid approach to the creation of focused domain knowledge that approximate an information seeker’s intent and context of inquiry, using top-down concept identification and bottom-up fact extraction. The domain model extraction framework builds models representing a domain or a context based on keyword descriptions. It first extracts a concept hierarchy from a general conceptual corpus that delineates the domain of interest and then connects its concepts with automatically extracted facts from specific textual sources to facilitate contextual browsing or faceted exploration of the search space. This work draws from, contrasts and extends research in both Ontology Learning and Information Extraction.

An ontology, often defined as “formal, explicit specification of a shared conceptualization” [6, 18], needs to formally define a domain in terms of concepts and relationships. It should also reflect an agreement among those who hold knowledge in the domain and who want to use the ontology as a means to communicate thoughts and facts about the domain. In the past, research on Ontology Learning has often assumed that no prior formal domain knowledge is available, so both concepts and relationships were extracted bottom-up from free text. Because of ambiguities and reference problems in natural language, this approach runs contrary to the goal of Ontology Learning. Ontologies should, by definition, contain rigidly defined concepts, rather than formal representations of concept mentions. Furthermore, correctness and consistency is required in ontologies but cannot be guaranteed in automated methods.

Some of the strict requirements for ontology learning can be relaxed for the purpose of creating domain models for IR applications [19]. A conceptual model that improves search and/or classification results is beneficial, even if it contains some concepts or facts that arguably do not belong to the domain of interest and thus may cause some filtering mistakes. Because of the potential errors in automatically created models, we will refrain from using the term ontology in this paper and use the term Domain Model instead, which we define as a conceptual description of a domain of interest that formally defines, disambiguates and grounds its concepts, but does not necessarily adhere to the strict requirements of absolute correctness and consistency. Maintaining
concept integrity is especially important when the created models are used as input for applications that need to interpret the extracted facts rather than directly present them to users. Examples of such applications are faceted browsing [52], Semantic Trailblazing [37] and content classification [11, 29, 39].

In this work, concept integrity is guaranteed by extracting a Domain Definition from a community-created or peer-reviewed conceptual corpus. The term Domain Definition refers to the conceptual basis of a model, expressing what concepts and entities exist in a domain and by which types of relationships they can generally be related, but not how they are actually related. This definition is similar to, but not the same as the definition of T-Box in the Semantic Web literature, because it includes individuals, whereas the T-Box only contains classes, properties and their logical restrictions.

When information sources are chosen for a top-down hierarchy extraction that take the social character of knowledge aggregation into consideration, broad agreement is practically built into the model creation. We can thus extract domain models that, despite their focus on an individual’s interest, represent a shared understanding of the domain knowledge, and fulfill the requirement of agreement [18]. Recent research in the area of Web 2.0 emphasized how communal creation of information organically results in common designations of concepts, for example in Wikipedia [22] or social bookmarking sites [20]. Projects that take advantage of these community-assigned designators to create more formal representations include DBpedia [4], in particular the DBpedia Infobox dataset, which presents Wikipedia Infobox knowledge in RDF (Resource Description Framework) format. Corpora that contain peer-reviewed (i.e. community-vetted) concept descriptions and are thus useful for the extraction of scientific knowledge include UMLS (Unified Medical Language System) [27].

Wikipedia/DBpedia Infoboxes and UMLS provide a large coverage of their domains in terms of concepts and relationship types, but they are very sparsely populated with facts that instantiate the relationship types. For example, the DBpedia 3.6 Infobox dataset has approximately 7 million asserted facts involving named object properties other than category membership and type. Given that Wikipedia currently has about 4 million articles that describe concepts, this means that every resource has on average fewer than two relationships with other resources. Therefore it is necessary to have a bottom-up approach to automated fact extraction from free text. Since attributes and relationships describe or explain the concepts that were extracted during Domain Definition, this step will be referred to as Domain Description.

The approach to Domain Description taken in this work is that of distantly supervised, concept-centric pattern-based IE. It is distantly supervised [31], because we do not assume the availability of annotated corpora. Rather, free text is analyzed for potential occurrences of known facts that are taken from the DBpedia Infobox and UMLS corpora. Due to the lack of annotated text corpora, the algorithm also operates on positive training examples only. It is concept-centric, because we are only looking for terms that have a high probability of identifying the concepts that were found during the Domain Definition step. Lastly, we developed a pattern-based IE algorithm that allows us to extract facts without performing expensive NLP methods, such as POS-tagging or parsing.

Most IE algorithms implicitly look at the types of relationships that are extracted as distinct. However, relationships can be arranged in a hierarchy, where a relationship can entail another one or the relationships can otherwise be semantically overlapping. This overlap can happen by design or because community-created corpora (e.g. Wikipedia) have different users define new relationships, even though a similar one already existed or because corpora that were created from multiple sources (e.g. UMLS) adopt overlapping descriptions from their sources. In a distantly supervised system, the types of relationships usually cannot be manually reviewed. Therefore it was crucial to find a statistical measure that can handle semantic overlap during extraction. The pertinence measure for relationships that is described in Section 4 leverages the textual grounding of relationship types as well as their overlap frequencies in the training data to determine extensional and intensional overlap. Extensional overlap happens when two relationships share the same subject-object pairs regardless of their semantics (e.g. person lives_in place; person died_in place). Intensional overlap is given when two relationships have similar meanings, such as part_of and physical_part_of. When extensional overlap is not recognized and avoided, it can distort the textual grounding of relationships. Relationships that overlap intentionally should not compete for the extraction result, but should rather all be considered as candidate results.

To give an idea of the system’s utility in creating models for scientific domains, we will show the development of a model created for the area of human cognitive performance. The model provides browsing background knowledge to Scooner [8], a browser that allows information retrieval along semantic trails. An excerpt of the model is shown in Figure 2, a more detailed analysis of the development of this model is given in Section 6.2.2.

The contributions of this work are:

1. A domain hierarchy extractor that creates a Domain Definition from a conceptual corpus, such as Wikipedia.
2. A concept-centric, distantly supervised relational-targeting IE algorithm using surface patterns occurring in free text for the extraction of the Domain Description.
3. A statistical pertinence measure that facilitates dealing
4. Recall enhancement using a pattern generalization algorithm

5. Extensive qualitative and quantitative evaluation of the Domain Description part, including analysis of the extracted patterns.

The remainder of the paper is structured as follows. Section 2 gives a broad overview over the steps involved in the domain model creation. The Domain Definition creation has previously been described in the first Doozer system\(^2\) [46]. To have a self-contained description of the work, this paper contains an abridged description of these prior contributions in Section 3 putting the new work on Domain Description in Section 4 in context. Section 5 discusses the combination of hierarchy creation and IE. In Section 6, the approach is evaluated extensively using manual and automated qualitative and quantitative methods. Section 7 places this work in the context of related work and familiar approaches and Section 8 concludes the paper.

2. OVERVIEW

In this section we summarize conceptual considerations that went into the 2-step approach of (a) Domain Definition and (b) Domain Description to model creation. We will discuss epistemological, semantic and NLP-related issues involved in model creation as well as the necessary assumptions we make about the data in order to overcome these issues.

The terminology used in this paper will mostly be familiar, however, we want to define some terms here to ensure a consistent understanding:

\(^2\)The Doozer hierarchy creation can be tested at http://knoesis-hpco.cs.wright.edu/DoozerServlet/

<table>
<thead>
<tr>
<th>Table 1: Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Doozer</strong></td>
</tr>
<tr>
<td><strong>Doozer++</strong></td>
</tr>
<tr>
<td><strong>concept</strong></td>
</tr>
<tr>
<td><strong>term</strong></td>
</tr>
<tr>
<td><strong>entity</strong></td>
</tr>
<tr>
<td><strong>class</strong></td>
</tr>
<tr>
<td><strong>category</strong></td>
</tr>
</tbody>
</table>

2.1 Epistemological Considerations

In social epistemology, it is believed that there is a general effort towards making truthful statements [17]. This means that we can generally trust a majority of independently made statements. However, it is clear that also many incorrect statements are made and perpetuated. These need to be identified or at least disregarded. The Wisdom-of-the-crowds paradigm states that a large number of people are able to solve difficult problems, as long as they are independent, given a good infrastructure and their answers are aggregated in an intelligent manner [45].

Applying this maxim to IE tasks, we should trust an assertion only when we find it asserted by various independent sources or when the source is created in a collaborative or peer-reviewed manner. In the Read-the-Web project, the term Macro-Reading was coined for this kind of accumulative reading of statements [9]. It assures that the extracted domain models, despite their focus on an individual’s inter-
est, represent a shared understanding of the knowledge that underlies the model creation.

There is usually general agreement about the existence of a concept, an event or an entity. The disagreement tends to appear with the description of the concept and by relating it to other concepts. Hence, we can make the safe assumption that concepts can be harvested from encyclopedias, ontologies, taxonomies or other concept repositories, whereas the facts describing the concepts should be extracted from free text in a manner that aggregates multiple mentions of each relationship. Accordingly, we will split the task of domain model creation into the two parts of Domain Definition and Domain Description.

2.2 Domain Model Creation

The conceptual separation of Domain Definition and Description is reflected in the use of different techniques for both tasks. When the domain is defined by extracting known concepts from a corpus that has clearly delineated concepts with unique identifiers, ambiguities can be avoided and the description step can work on the basis of concepts, rather than just terms.

To summarize the conceptual considerations, this work is built on the following premises:

1. Humans succeed at identifying, defining and describing concepts.
2. Semantic Domain Models represent a human conceptualization and abstraction of the world.
3. Unambiguous (semi-)formal concept identifiers are available in greater abundance than concept descriptors.
   (a) Encyclopedias, glossaries and vocabularies provide concept designators.
   (b) Community-created or peer-reviewed Encyclopedias, glossaries and vocabularies express a shared view of a domain.
⇒ Extract domain definition top-down from such corpora.
4. Concept descriptions, such as attributes and relationships are plentiful in informal text and manifested in multiple documents.
5. An aggregation of multiple statements about a concept yield a more accurate description.
   (a) A macro-reading-based [9] IE approach aggregates distributed information
   (b) Pattern-based IE inherently conforms to an aggregative process
⇒ Extract domain description bottom-up from free text

2.2.1 Domain Definition

Domain Definition is accomplished by restricting existing structured or semi-structured sources to only contain concepts pertinent to a focus domain. In this work we use Wikipedia as a knowledge source. We make the assumption that most concepts and entities of interest are represented by articles in Wikipedia and concept labels are represented by article titles as well as anchor texts that link to the articles.

Over the years, Wikipedia has become a high-quality encyclopedia. With an ever growing number of articles, Wikipedia covers an impressive number of concepts of general interest. A Nature article from 2005 found that the amount of factual errors in Wikipedia is not significantly different from those in the Encyclopedia Britannica [16]. In previous work [47] we demonstrated how most articles on Wikipedia mature over time and converge to a stable state.

Even more than the actual content of the articles, the naming and classification of concepts is of particular interest to the hierarchy creation part of this work. It has been shown that the URLs of Wikipedia articles are valuable as general concept identifiers in ontologies [22], because they have been community-vetted and are unambiguous.

A valid criticism of using Wikipedia as a source of conceptual knowledge is that Wikipedia does not provide enough depth of coverage for highly specialized domain models. However, we believe that the benefit of having concept integrity outweighs the lack of recall in some generated models.

2.2.2 Domain Description

Many approaches to IE extract assertions from text by parsing sentences and promoting syntactic Subject-Predicate-Object structures to semantic assertions (triples). Open IE approaches of this kind have been termed Structural Targeting Open IE [2, 51]. The problem with elevating single syntactic structures to semantic assertions is that concept integrity is not guaranteed, therefore the extracted facts cannot be mapped to a formal model and hence fail to refer to real-world entities. In addition to the reference problem, when an assertion is extracted from a single phrase found in a single document, there is no notion of a shared understanding and no guarantee of correctness. Some extractors, such as Textrunner [2], combat this problem by ranking extracted assertions by the number of occurrences found. A further problem is that in natural language use, there is a many-to-many relation between verbs and formal relationships due to synonymy and polysemy. For example, the predicate $X$ causes $Y$ could be expressed in the phrases “$X$ causes $Y$”, “$X$ induces $Y$”, “$Y$ is caused by $X$”, and many more. On the other hand, the polysemous verb “break”, for example, can be used in the senses separate, interrupt, violate or destroy. Without further disambiguation, Structural Targeting Open IE is likely to interpret the verb “break” as having the same meaning in the sentences “Marcus broke the glass” and “Marcus broke the law”. Finally, some relationships are rarely expressed as
verbs. For example, we do not say or write “Ottawa capitalizes Canada”. Rather, we will find phrases such as “A magnitude 5.5 earthquake recorded near the Canadian capital of Ottawa has rattled some nerves in the Chicago area.”

We address the ambiguity problem by using supervised pattern-based extraction rather than NLP-based extraction. Relationship types are manifested in many different patterns across a corpus. The domain description algorithm learns these patterns by finding examples for triples from a fact corpus in text and then generalizing the patterns found for single facts to patterns for types of relationships. Fact corpora are widely available on the Linked Open Data (LoD) cloud [3]. This kind of IE has been termed Relational-Targeting IE [51]. In our case the openness of the algorithm stems from the openness of the fact corpora on LoD, which are continuously evolving and the algorithm continuously trains on the types of relationships that emerge. Hence we refer to our approach as Semi-Open.

Semi-Open Relational-Targeting IE resolves relational ambiguities in the training phase by accumulating different patterns for a semantic relationship. The problem that a pattern such as “X broke Y” can indicate different semantic relations still persists. However, thanks to the training procedure, we have an idea how likely it is that it indicates for example violate or destroy. Evidence for a relationship is not just taken from one occurrence of a pattern, but the accumulated evidence of multiple patterns between the surface representations of two concepts. Whereas a single occurrence of “Marcus broke the glass” ambiguous for the extractor, the accumulated evidence of “Marcus broke the glass”, “Marcus shattered the glass” and “Marcus destroyed the glass” disambiguates the relationship between Marcus and the glass.

In many cases, structural targeting open IE also does not enforce the presence of meaningful concept or entity descriptors in the subject or object of the extracted statements. Many of the statements extracted by ReVerb [14] or WOE [51], for example, contain pronouns (e.g. you, she, we, her, it, etc.), or phrases that are only referring to concepts in context (e.g. “six of 11 countries” or “the legislation”). Not only does this result in meaningless extracted statements, it can also lead to adding incorrect statements to proper concepts. For example, the noun phrase “the legislation” was extracted as a subject in the triple “the legislation achieved → a number of needed reforms”. In the original sentence, “the legislation” is a synecdoche (i.e. a deferred reference) for a particular legislation that was probably introduced in a previous sentence. However, taken in isolation, the triple may be seen as referring to the concept of legislation in general. This makes obvious the need for a concept-centric extraction to ensure the validity and proper reference of a triple’s subject and object. This not only assures correct-ness, but also allows mapping of extracted facts into existing formal models.

Relational Overlap: Many types of relationships are semantically overlapping or one type of relationship is entailed by another. The overlap can be intensional or extensional.

Intensional overlap is given when all instances of a subproperty are necessarily instances of the superproperty, because the definition of the subproperty entails the definition of the super property. For example, the relationship physical part of entails part of.

Extensional overlap is given when the instances of two relationships overlap without necessarily belonging to both. For example, the relationships birthplace and deathplace often share common subject-object pairs, because for example a person was born, lived and died in the same place. Extensional similarity is given when the instances of relationships happen to overlap, but the definitions of the relationships are different.

In order to improve extraction in light of intensional or extensional overlap, the algorithm should be able to find distinctive features of extensionally overlapping relationships, even though many training examples are overlapping and thus the pattern manifestation of the relationships is similar. It also needs to find features that intensionally overlapping relationships share, so they are not seen as discriminative features between these relationships. These classification problems inherent to OpenIE are addressed by the development of a semantic pertinence measure that balances the impact of intensional and extensional overlap based on both formal and textual representations of relationships.

Each relationship is represented by the collection of patterns that were found as the manifestation of the relationship in text. In particular, each pattern is assigned a probability of being an indicator of a relationship. Thus, a vector space representation of these probabilities is an intuitive choice. In such a representation, a relationship is expressed as a vector of pattern probabilities and the collection of relationships is expressed in a relationship-to-pattern(R2P) probability matrix. Vector space representations have been proven to be successful in document classification and relationships extraction [49]. Patterns have proven effective in identifying relationships [1, 48]. Not only can expensive POS tagging and parsing steps be skipped, but index representations of textual data in the form of term vectors and position vectors can directly be used. Thus, patterns and entities can be identified in the index, rather than having to analyze the full text.

Based on these considerations Domain Description is defined within the following technical framework:

• Pattern-based representation of relationship types

3An evaluation of triples extracted by TextRunner, WOE and Reverb can be found at http://www.cs.washington.edu/homes/afader/data/reverb_emnlp2011_data.tar.gz
Figure 3: Steps 1(Full-text search) and (Semantic Similarity) in the expansion process

- Vector-space model for the representation of relationships as pattern-occurrence probabilities
- Probabilistic multi-class classifier for fact extraction
- Pertinence-based computation of pattern importance

3. DOMAIN DEFINITION

This section describes the creation of domain hierarchies as the first step toward domain model creation. Based on the earlier discussion about concept integrity and reference, the approach taken in this work is to extract a domain hierarchy from a larger collection of concepts, namely from the Wikipedia article and category hierarchy. To make this process user-friendly and applicable to the task of on-demand creation of domain hierarchies, it was imperative to devise a system that intelligently selects the concepts that are pertinent to a domain solely based on a partial keyword description. A user may only know some of the concepts that are important in a domain, but she still wants a fairly complete, yet focused description of her domain of interest.

The overall process of creating a comprehensive domain hierarchy from a simple set of keywords follows an Expand and Reduce paradigm that allows the system to first explore and exploit the concept space before reducing the concepts that were initially deemed interesting to those that are most important and most significant to the domain of interest. Table 2 shows the symbols that will be used throughout this section.

**Table 2: Terminology**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Set of concepts without categories</td>
</tr>
<tr>
<td>C</td>
<td>Concept</td>
</tr>
<tr>
<td>C&lt;sub&gt;article&lt;/sub&gt;</td>
<td>Concept described by a Wikipedia article</td>
</tr>
<tr>
<td>T</td>
<td>Category hierarchy</td>
</tr>
<tr>
<td>Dom</td>
<td>Domain: The topic of the model</td>
</tr>
<tr>
<td>M</td>
<td>Domain Model including individual concepts and categories</td>
</tr>
<tr>
<td>W</td>
<td>All articles on Wikipedia</td>
</tr>
<tr>
<td>G&lt;sub&gt;W&lt;/sub&gt;</td>
<td>Graph structure of Wikipedia articles and Categories</td>
</tr>
</tbody>
</table>

3.1 Expansion

In the expansion steps (Figure 3), recall is maximized to allow as many concepts as possible to be taken into account while maintaining a sensible focus on the domain of interest.

**Full Text Search — Exploring the knowledge space:** For a full-text search over the content of the Wikipedia articles, the complete set of articles is indexed using the Apache Lucene search engine. The set of concepts $C_{search}$ contains the concepts (i.e., the URIs) of those indexed Wikipedia article that match a query with a score greater than a given threshold $\epsilon_{search}$ and/or smaller than a given maximum search rank (Equation 1).

$$C_{search}(query) = \{ C_{article}, article \in hits(query) \} \quad \text{score(article) > } \epsilon_{search} \lor \text{rank(article) < maxRank} \quad (1)$$

**Graph-Based Expansion — Exploiting the knowledge space:** The graph-based expansion returns articles that are semantically related to the initial search results. Semantic similarity is computed using a weighted common neighbors metric [28] over Wikipedia article links. The semantic similarity between nodes $a$ and $b$ is defined as the sum of the average weights of their shared neighbors, normalized by the average of the node degrees. Let $M$ be the adjacency matrix of Wikipedia. Let $N(a)$ stand for the neighborhood of $a$ and $w(N(a))$ stand for the link weights to $a$'s neighboring nodes, and includes all the articles that link to or are linked to $a$. Thereby, $w(N_i(a))$ describes the link weight to the $i$th neighbor of $a$. The weights depend on the kind of link between $a$ and each of its neighbors, as shown in Figure 4. Equation 2 is similar to the first iteration of SimRank [23]; the difference is in the normalization factor and the weighted links.
based expansions (Equation 4).

\[ C_{\text{base}} = C_{\text{search}} \cup C_{\text{sim}} \] (4)

### 3.2 Reduction

Whereas the expansion steps are used to gather knowledge in a recall-oriented way, the reduction steps increase precision and reduce the set of concepts to concisely match the domain of interest.

**Probability-based reduction:** For each concept in the set of extracted concepts \( C_{\text{base}} \), we compute two relevance probabilities with respect to the broader domain of interest. Equations 5 and 6 show these conditional probability computations. Equation 5 indicates the importance of a concept for the domain. A probability of 1.0, for example, would indicate that every time the concept appears, it is within the domain of interest. Equation 6 shows the inverse: how commonly used is the concept in the domain? Knowing both measures is not only important for this pruning step, but also for a potential use of the created domain model in probabilistic document classification tasks. If the importance of a concept is less than one of the predefined thresholds \( \epsilon_1, \epsilon_2 \), it is discarded from the set of domain concepts \( C_{\text{base}} \), resulting in the reduced set of domain concepts \( C_{\text{red}} \) (Equation 7).

Naturally, concepts are represented in text using terms. To determine the conditional probabilities, the terms that were identified as denoting a concept are used to query the Wikipedia text, where the query takes the following form: 

\[
\text{query}(Q) = \text{search}_{\text{index}}(\{\text{term} \mid \text{denotes(term, q), } q \in Q\})
\]

where \( Q \) indicates the set of concepts that is queried. The acquisition of terms that denote concepts is described in the next section (3.3). The whole domain \( \text{Dom} \) is queried for by performing a disjunctive query over all concepts that were identified in \( C_{\text{base}} \). Since the result of this query is static, it can be executed once per model and then stored, so querying for a large amount of concepts does not significantly influence the efficiency of the algorithm.

\[
p(\text{Dom}|C) = \frac{|\text{query(\text{Dom} \cap C)}|}{|\text{query(C)}|}
\] (5)

\[
p(C|\text{Dom}) = \frac{|\text{query(\text{Dom} \cap C)}|}{|\text{query(Dom)}|}
\] (6)

\[
C_{\text{red}}(\text{Dom}) = \{C \in C_{\text{base}} \mid p(\text{Dom}|C) \geq \epsilon_1 \\
\land p(C|\text{Dom}) \geq \epsilon_2\}
\] (7)

**Building a category hierarchy** is not only essential to organize the individual concepts, it also provides better means for further pruning. The initial category hierarchy is built by iteratively adding layers of super-categories over the individuals in \( C_{\text{red}} \) until the most specific common super-category is found, resulting in the category hierarchy \( T \).

Finally, the domain model \( M \) is constructed as the union of the set of remaining concepts \( C_{\text{red}} \) and the imposed category hierarchy \( T \):

\[
M = C_{\text{red}} \cup T
\] (8)

**Depth Reduction:** In many cases, deep linear branches of categories remain as artifacts of the category building. Unpopulated and unbranched category hierarchies can be collapsed without loss of relevant knowledge, because many categories on Wikipedia are organizationally, but not ontologically relevant. Categories of the form \( X \text{ by } Y \), such as People by Occupation, People by nationality usually do not have direct category members and can easily be collapsed without losing ontological knowledge. This reduction in depth and the resulting increase in fan-out makes the domain model user-friendlier.

### 3.3 Synonym Acquisition

Since the extraction of domain models is concept-based, rather than term-based, it is necessary to find alternative notations or labels for the concepts in the hierarchy. The Wikipedia article names are unambiguous identifiers and as such not necessarily of the form we are used to when talking about the concept described by the article. A domain model that is used for IE or text classification needs to contain these identifiers and synonyms for the concept of the article. The Wikipedia article for the capital of the United States, for example has the Wikipedia name “Washington, D.C.” as a unique identifier. We expect to find different identifiers in text, though, such as “Washington” or just “D.C.”. Anchor texts for links have proven to be a good indicator for alternative concept labels [7]. This is made easier on Wikipedia than on the Web in general, because Wikipedia associates a clear concept with the link. The probability that a term is a synonym of an Article name and hence an identifier of the concept described by the article is given by Equation 9, the conditional probability that a term links to an unambiguous article:

\[
p_{\text{syn}}(\text{Term}, C) = \frac{|\text{links}_{\text{to}}(\text{Term, Article}_C)|}{\sum_{a \in \text{W}} |\text{links}_{\text{to}}(\text{Term, a})|}
\] (9)
In the OWL-serialized domain hierarchy, these synonyms are represented as labels as well as individual values of annotation properties that contain both the term and \( p_{syn} \), so the synonyms can be used in classification.

The next section will discuss the embellishment of the extracted concept hierarchy with fact triples using named relationships.

4. DOMAIN DESCRIPTION

Domain description is the second step of domain model creation. As the name suggests, it is concerned with describing the concepts that were found to be relevant for the domain. This description is achieved by connecting the concepts or individual entities in the domain with named relationships that indicate how one concept or entity relates to another. The relationship types are defined in an implicit or explicit schema, for example in DBpedia[4] or UMLS[27]. As stated in the overview section, the factual quality of the extraction, apart from the performance of the algorithm, is guaranteed by either extracting from community-created or peer-reviewed corpora, such as Wikipedia and MEDLINE\(^5\), or by aggregation of evidence for each fact in the case of uncontrolled corpora, such as general Web pages.

A fact is an instantiation of a named relationship, involving a subject and an object concept. In order to be called a fact, it also needs to express a true real-world connection. An extracted fact is thus a verified extracted statement. Since this work is only concerned with binary relationships, all the statements will formally be represented as \( \langle \text{Subject}, \text{Relationship}, \text{Object} \rangle \) triples. The triple representation follows the RDF standard that is prevalent in the Semantic Web. Many datasets on the Linked open Data (LoD) cloud are also represented in this format. However, since class membership can be expressed by a binary relationship, this can also be extracted.

Fact extraction is implemented as a classification of concept pairs into relationship types. Since each concept pair can potentially participate in multiple relationships (e.g. birthplace and residence), a probabilistic multi-class relationship classifier was developed that operates on a sparse concept-pair to pattern vector space representation. One impediment that has troubled purely pattern-based approaches to IE is low recall. This was improved by using a pattern-generalization step. Another challenge faced was that of classifying into multiple classes of relationships where appropriate. This was addressed using a statistical pertinence measure that reevaluates the importance of a pattern to a relationship class based on the similarity to other classes.

Using only positive examples of features and having only positive examples of relationship types, the classifier must face the problem of missing discriminating data. Still, with the large and ever-growing LoD datasets at hand, the number of relationship types that the pattern occurrences are classified into also grows and thus our ability to discriminate between positive examples. The challenge is to discriminate relationships that seem very similar to the classifier because of the above mentioned problems. The main goal of a positive-only classifier is thus emphasizing differences and penalizing similarities between different relationships, while not penalizing similarities between similar relationships. One unique feature of this classifier is thus an increase in discriminative performance as the number of relationships that are classified grows. This is a significant improvement over traditional supervised IE techniques that tend to have better performance when only a limited number of classes are trained on.

This section is structured as follows. First, it will introduce the notion of surface patterns. Then it will give a general idea of the probabilistic framework, before showing how this is represented in a vector space. Subsequently it will explain the pertinence measure and show examples how this measure impacts the importance of individual patterns. Finally it will show how the classifier is built and applied.

4.1 Surface Patterns

The pattern-based IE algorithm developed for this work uses distant supervision [31] to extract patterns that express formal relationships in free text. When a phrase is found in text that potentially expresses a fact from the training corpus, it is generalized into a pattern by replacing the terms that indicate the subject/object pair of the fact with placeholders. Hence, a pattern is of the form \( \langle \text{Prefix} \rangle \langle L_1 \rangle \langle \text{Infix} \rangle \langle L_2 \rangle \langle \text{Postfix} \rangle \) with \( \langle L_1, L_2 \rangle \) indicating a pair of concept labels found using Equation 9 that denote subject and object concepts of the triple. The textual grounding of a triple is the collection of phrases that express the semantic content of the formal statement.

As an example consider the triple \( \langle \text{Albert Einstein, birthPlace, Ulm} \rangle \) and potential textual manifestations of this fact. The phrase “Albert Einstein was born in Ulm” is added to the pattern dictionary as \( P_1 := \langle \text{Subject} \rangle \). Similarly, the phrase “Ulm is the birthplace of Albert Einstein” is represented as \( P_2 := \langle \text{Object} \rangle \). Assuming that these patterns occur in the frequencies \( f_i \) and \( f_j \) respectively, the triple \( \langle \text{Albert Einstein, birthPlace, Ulm} \rangle \) is then grounded by the textual occurrences \( \langle P_1, f_i \rangle \) and \( \langle P_2, f_j \rangle \).

The top-down step to domain definition gives us the probabilities with which \( L_1 \) and \( L_2 \) denote the proper concepts mentioned in the triple, because concept labels and the probability that they actually refer to the concept are extracted along with the concepts. Section 4.2 will show how these probabilities are used.

4.1.1 Pattern Generalization

A major drawback of surface patterns is their lack of variability and associated low recall. Apart from such widely
4.2 Probabilistic Framework

A probabilistic classifier for relationships intuitively answers the question: “Which relationship is likely expressed when two entities appear with these patterns?” It is also manually verifiable, which makes it a good candidate for a prototype application.

Before discussing solutions to this problem the general idea of the probabilistic classification is outlined.

### Table 4: Terminology

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(C)</td>
<td>Concept</td>
</tr>
<tr>
<td>(C)</td>
<td>Set of all concepts</td>
</tr>
<tr>
<td>(S, O)</td>
<td>Subject/Object concept of the triple</td>
</tr>
<tr>
<td>(L)</td>
<td>Term/Label</td>
</tr>
<tr>
<td>(L_S, L_O)</td>
<td>Term expressing the subject/object of a triple</td>
</tr>
<tr>
<td>(T_S, T_O)</td>
<td>Semantic type/class of the subject/object</td>
</tr>
<tr>
<td>(P)</td>
<td>Surface pattern</td>
</tr>
<tr>
<td>(R)</td>
<td>Relationship type</td>
</tr>
<tr>
<td>(\mathcal{R})</td>
<td>Set of all relationship types</td>
</tr>
<tr>
<td>(R2P)</td>
<td>Relationship-Pattern matrix</td>
</tr>
<tr>
<td>(CP2P)</td>
<td>Concept-Pair-Pattern matrix</td>
</tr>
<tr>
<td>(CP2R)</td>
<td>Concept-Pair-Relationship matrix</td>
</tr>
<tr>
<td>(M_D, M_R)</td>
<td>Domain and Range prior probability matrices</td>
</tr>
</tbody>
</table>

**Derivation of the Classifier** The following paragraphs will show the derivation of the probabilistic extraction framework. IE is hereby cast as a classification from concept pairs into relationship types using surface patterns as features. The terminology used in the derivation is given in Table 4.

It is important to observe that a distinction is made between the concepts that participates in the relationship and the labels that denote the concepts in text. Similarly, the extraction algorithm operates on concepts rather than labels. However, since concepts do not actually appear in text, the grounding of the concepts in text has to be done using the denoting labels. Given the ambiguity in mapping, a single occurrence of a term pair in text is not enough to indicate the concept pair that is sought. Similarly to the mapping of pattern occurrences to types of relationships, the mapping of multiple occurrences different terms to one concept assure the proper references.

Suppose a given concept pair \((S, O)\) from the set of all concepts on Wikipedia \(C\) that should be classified into one or more relationship types \(R\). The general goal is hence to find a solution to the conditional probability \(p(R|S, O)\) and identify all relationships \(\mathcal{R}_{S,O}\) that have a sufficiently high confidence of relating \(S\) to \(O\) (Equation 11).

\[
\mathcal{R}_{S,O} = \{ R \in \mathcal{R} | p(R|S, O) > \epsilon_{rel} \} 
\]  

(11)

The features of the classifier are the patterns found in text that are potential manifestations of the concept pair and the relationship as described in section 4.1. Free text, however, will contain ambiguous terms that denote \(S\) and \(O\). Section 3.3, explained how to find probabilities for synonyms of ar-

---

Table 3: Pattern-generalization example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((\text{Subject})) graduated in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>2</td>
<td>((\text{Subject})) graduated in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>3</td>
<td>((\text{subject})) graduated in * from ((\text{object}))</td>
</tr>
<tr>
<td>4</td>
<td>((\text{Subject})) graduated in * from ((\text{Object}))</td>
</tr>
<tr>
<td>5</td>
<td>((\text{Subject})) graduated * in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>6</td>
<td>((\text{Subject})) graduated * in 1998 * from ((\text{Object}))</td>
</tr>
<tr>
<td>7</td>
<td>((\text{Subject})) graduated * * from ((\text{Object}))</td>
</tr>
<tr>
<td>8</td>
<td>((\text{Subject})) graduated * * * from ((\text{Object}))</td>
</tr>
<tr>
<td>9</td>
<td>((\text{Subject})) in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>10</td>
<td>((\text{Subject})) * in 1998 * from ((\text{Object}))</td>
</tr>
<tr>
<td>11</td>
<td>((\text{Subject})) * in 1998 * from ((\text{Object}))</td>
</tr>
<tr>
<td>12</td>
<td>((\text{Subject})) * in * * from ((\text{Object}))</td>
</tr>
<tr>
<td>13</td>
<td>((\text{Subject})) * * in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>14</td>
<td>((\text{Subject})) * * * in 1998 from ((\text{Object}))</td>
</tr>
<tr>
<td>15</td>
<td>((\text{Subject})) * * * * * from ((\text{Object}))</td>
</tr>
<tr>
<td>16</td>
<td>((\text{Subject})) * * * * * from ((\text{Object}))</td>
</tr>
</tbody>
</table>

Applicable patterns as “\((\text{Subject})\) was born in \((\text{Object})\)”, there will be many very specific patterns that are only applicable to few concept pairs, such as “\((\text{Subject})\) graduated in 1998 from \((\text{Object})\)”. An ideal generalization of this pattern to indicate an \textit{almaMater} relationship is “\((\text{Subject})\) graduated in \* from \((\text{Object})\)”. This is arguably the best generalization of the original pattern. However, pattern extraction is agnostic to the semantics or even the part of speech or sentence position of the generalized tokens and hence the generalization is also unaware of the semantics of the tokens that are generalized. Since no parse tree exists, the kinds of generalization that are used in NLP-based approaches, e.g. [42], where shortest paths through parse trees are used to generalize patterns can also not be used as a guide. For this reason, all possible generalizations have to be created and then evaluated with respect to their predictive power for a relationship. Patterns with low predictive power will be pruned. After pruning, the set of remaining generalized patterns approximates the kind of generalization that is done in NLP-based generalization.

Table 3 shows an example of a complete generalization, where an exponential \(2^{\text{tokens}}/2\) patterns are produced. To save space an example pattern with a 4-token infix pattern was chosen, leaving out prefix and postfix. In practice, most of the patterns that contain more than 2 wildcards are pruned during minimization, because they are insignificant indicators for specific relationships. Thus the application usually limits the generalization to \(\lfloor \text{tokens}/2 \rfloor \) wildcards per pattern, resulting in \(\hat{n}\) patterns. In Table 3, this corresponds to rows 1-7, 9-11 and 13. Equation 10 computes the number of generalized patterns that are created from a raw pattern, when the number of wildcards is restricted, leading to a slower growth in the number of patterns.

\[
\hat{n} = \left( \frac{|\text{tokens}| + |\text{wildcards}| - 1}{|\text{wildcards}|} \right) 
\]  

(10)
A factor that helps in classification, but is dependent on the availability of background knowledge, is the information on domain and range of relationships, expressed here as the probability $p(R|T_S, T_O)$ of seeing a relationship given a domain $T_S$ and a range $T_O$. Here, as well, the joint probabilities are too restrictive for an open extraction so we assume independence of $p(R|T_S)$ and $p(R|T_O)$.

Depending on the background knowledge that is present in the form of ontologies, taxonomies and dictionaries, the terms $p(S|L_S), p(O|L_O), p(R|T_S)$ and $p(R|T_O)$ may be only partially available or not at all. In this case the classifier operates purely on the language model, rather than on the combined semantic model/language model. In a well-designed ontology, relationships will be assigned domains and ranges. However, in the case of community-created sources such as DBpedia this is more difficult. Even though an ontology exists that covers the entities on DBpedia, it is too coarse-grained to properly match the models produced in the Domain Definition step and also does not assign domain- and range restrictions to all properties. In this case, the probabilities $p(R|T_S)$ and $p(R|T_O)$ can be derived bottom-up, by analyzing the category-coverage of the facts in the KB as shown in Section 4.5.

The third and central component of the probabilistic framework are the relationship-pattern probabilities $p(R|P)$, i.e. the probability of seeing a relationship in the presence of a specific pattern or a vector of relationship probabilities given a vector of pattern frequencies. Separating $p(R|P)$ allows us to build a fixed pattern representation for relationships.

Figure 5 depicts a Bayesian Network that graphically models the classifier, showing how it operates on a Semantic model and a statistical language model in a unified manner. The probability $p(R, S, O)$ of a relationship occurring with a subject and an object can be rewritten as $p(R, S, O, P, L_S, L_O, T_S, T_O)$, which, based on the Bayesian Network, is formalized in Equation 12. The equation makes use of the independence assumption of $p(S|L_S), p(O|L_O), p(R|T_S)$, and $p(R|T_O)$ and approximates $p(R|T_S, P, T_O)$ as the product of $p(R|T_S), p(R|T_O)$ and $p(R|P)$. Specifically, as the probability of the presence of a relationship $R$, between $S$ and $O$ is computed over all the patterns that $L_S$ and $L_O$ appear in, the classifier sums over the probabilities of all occurrences of a pattern with $S$ and $O$, each weighted by the probability that its pattern indicates the relationship $R$. For the types, the probability is maximized over the domain and range types, indicating that the $T_S$ and $T_O$ form hierarchies and the type that has the strongest support as a domain or range for $R$ is chosen.

$$p(R, S, O) \approx \sum_{L_S \in S} \sum_{L_O \in O} \sum_{P \in \text{docs}} \frac{p(P|L_S, L_O) \cdot p(R|P) \cdot p(S|L_S) \cdot p(O|L_O) \cdot \max_{t_S \in T_S} p(R|t_S) \cdot \max_{t_O \in T_O} p(R|t_O)}{p(R|P)}$$

The values for $p(R|P)$ are the most difficult to derive, because a distant supervision approach is used for training without access to negative training data to learn these probabilities. Also, as mentioned above, no a priory knowledge is assumed of the relationship semantics and thus the extent of the semantic overlap between relationships needs to be obtained during training. The next subsections describe the acquisition of $p(R|P)$ using distantly supervised training to create a

### 4.3 Vector-Space model

To get a vector-space representation of the probabilistic model that describes $p(R|P)$, a distantly supervised training procedure is first used to accumulate patterns for individual fact occurrences in a $(\text{Concept-Pair, Pattern frequency})$ matrix.
The row vectors represent the frequencies of the patterns in which a concept pair appears. During training these are the \((S, O)\) pairs found in LoD triples. The frequencies are accumulated into a \{Relationship, Pattern\} matrix \(R2P\) that can be seen as a language model for relationship mentions in text. In the application phase patterns between previously unseen concept pairs are compared to \(R2P\) to yield candidate relationship types the concept pair participates in. The following steps detail the matrix creation:

1. Find pattern representations of training facts in the text corpus as described in Section 4.1 using distant supervision [31]. For every textual manifestation of a fact, replace the terms denoting subject and object in the triple with \((Subject)\) and \((Object)\) placeholders to generate a pattern. If it is a new pattern, it is added to the pattern dictionary. The internal representation of a fact then becomes a vector that maps the Subject-Object concept pair to a vector of pattern frequencies. A Concept pair to pattern matrix \(CP2P^R\) combines the individual vectors (Equation 13). The superscript \(R\) indicates that in the training phase the vectors contain information about the relationship in the triple, so accumulated representations for relationships can be derived. The same procedure is however used during application, when only the concept pairs are known and the relationship needs to be extracted. The weighted frequency for the occurrence of a concept pair \(\langle S, O \rangle\), and a pattern \(P_j\) is defined as the product of the probability that a term-set pair \(\{LS|LS\ \text{label of } S\} :: \{LO|LO\ \text{label of } O\}\) indicates \(\langle S, O \rangle\), and the frequency of seeing the pattern with any of these term pairs.

\[
CP2P^R_{ij} = \text{weighted frequency}(\langle S, O \rangle_i, P_j) = \sum_{\langle LS, LO \rangle \in \langle S, O \rangle_i} (|P^L_{j, LO}| \cdot p(S|LS) \cdot p(O|LO)) \tag{13}
\]

2. Generalization - The coverage of patterns is increased by substituting tokens in the pattern with wildcard characters as described in Section 4.1.1. Usually a generalized pattern is derived from multiple original patterns. The frequencies of the generalized pattern is then computed by adding the frequencies of these original patterns. For example, if the frequency for \"\text{\langle Subject\rangle graduated in 1998 from \langle Object\rangle}\" is 5 and the frequency for \"\text{\langle Subject\rangle graduated in \langle Object\rangle}\" is 7, then the frequency of the generalized pattern \"\text{\langle Subject\rangle graduated in * from \langle Object\rangle}\" is 12.

3. To construct a matrix that contains the probabilities \(p(R|P)\) of seeing a relationship type when encountering a pattern, we first build a matrix that contains the probabilities \(p(P|R)\). This is done by adding all concept pair vectors in \(CP2P^R\) that indicate one relationship type into one row vector in \(R2P\) and then normalizing the row vectors. According to Equation 14, all vectors in \(CP2P^R\) that are annotated with the \(k^{th}\) relationship are added to row \(R2P_k\).

\[
R2P_{i}^{freq} = \sum_{i=1}^{n} CP2P^R_{ik} \tag{14}
\]

Additionally, the following frequency criteria need to be met:

- for each pattern, the overall weighted frequency is above a threshold \(t_1\)
- the number of distinct subject-object pairs that the pattern occurs with is above a threshold \(t_2\)

Taking both the raw frequency of the pattern and the frequency of the training facts that lead to the pattern into account reduces noise by ensuring that a pattern not only occurs often enough, but is also not specific to a particular concept pair.

The algorithm assumes a uniform distribution of relationship types in the world. Even though there is a nonuniform distribution of these types in a fact corpus, it is likely that this distribution is skewed. In the case of Wikipedia/DBpedia, for example, a majority of facts stem from sports-related descriptions. However, this distribution is skewed. With a potentially unbounded number of relationship types in reality, it is safe to assume that the prior probability for each individual type will eventually approach 0. Thus, the more relationship types are available for classification, the more realistic a uniform distribution becomes. For this reason the rows in the \(R2P\) matrix are normalized (Equation 15), before computing each field \(a_{ij}\) in \(R2P\) as the probability of seeing \(R_i\) when encountering \(P_j\) in text (Equation 16).

\[
R2P_{ij}^{norm} = \frac{R2P_{ij}^{freq}}{\sum_{k=1}^{n} R2P_{ik}^{freq}} \tag{15}
\]

\[
R2P_{ij} = \frac{p(P_j|R_i)}{\sum_{k=1}^{n} p(P_j|R_k)} \tag{16}
\]

4. Minimization : prune low probability patterns to reduce noise and to reduce the size of the matrix. The minimum frequency requirements in step (3) ensured that infrequent patterns that were too specific are not taken into consideration. Here, patterns that are too general to be of value for the classification are removed.

The resulting matrix \(R2P\) that maps relationships to their occurrence with specific patterns is basically a pattern-based language model designed to detect occurrences of relationships. It is of fairly low complexity, because instantiated patterns can be seen as 3-grams, consisting of subject designator, object designator and relationship designator.
Table 5: Extensional and Intensional similarity between relationships. The top half shows taxonomy relationships specific to the biology domain, the lower half shows domain-independent relationships. The first 2 columns show the relationship pair, the next 4 columns show extensional attributes with the number of shared subject-object pairs and the overall number of instantiating facts for each relationship type. Fraction\textsubscript{min} indicates the fraction of overlap measured by the relationship with the least number of instances. The next 3 columns indicate intensional similarity, computed using pertinence (+\textit{sim\textsubscript{int}}), omitting pertinence (−\textit{sim\textsubscript{int}}) and the difference in similarity. A positive value indicates that +\textit{sim\textsubscript{int}} assigned higher similarity than −\textit{sim\textsubscript{int}} and vice versa. The last column indicates \textit{sim\textsubscript{rel}}, the overall relational similarity, taking extensional and intensional similarity into account.

<table>
<thead>
<tr>
<th>Relationship Pair</th>
<th># Shared</th>
<th># Rel-1</th>
<th># Rel-2</th>
<th>Fraction\textsubscript{min}</th>
<th>+\textit{sim\textsubscript{int}}</th>
<th>−\textit{sim\textsubscript{int}}</th>
<th>Diff</th>
<th>\textit{sim\textsubscript{rel}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>family genus</td>
<td>603</td>
<td>310775</td>
<td>110358</td>
<td>0.005464</td>
<td>0.10496</td>
<td>0.09912</td>
<td>0.001304</td>
<td>0.10438</td>
</tr>
<tr>
<td>order class</td>
<td>391</td>
<td>184907</td>
<td>179328</td>
<td>0.002180</td>
<td>0.05780</td>
<td>0.04194</td>
<td>0.01586</td>
<td>0.05768</td>
</tr>
<tr>
<td>order family</td>
<td>812</td>
<td>184907</td>
<td>310775</td>
<td>0.004391</td>
<td>0.05089</td>
<td>0.03998</td>
<td>0.01090</td>
<td>0.05067</td>
</tr>
<tr>
<td>kingdom class</td>
<td>1</td>
<td>168222</td>
<td>179328</td>
<td>0.000006</td>
<td>0.03572</td>
<td>0.03784</td>
<td>−0.00211</td>
<td>0.03572</td>
</tr>
<tr>
<td>family class</td>
<td>228</td>
<td>310775</td>
<td>179328</td>
<td>0.001271</td>
<td>0.03114</td>
<td>0.03177</td>
<td>−0.00062</td>
<td>0.03110</td>
</tr>
<tr>
<td>phylum order</td>
<td>51</td>
<td>129784</td>
<td>184907</td>
<td>0.000393</td>
<td>0.00873</td>
<td>0.01234</td>
<td>−0.00360</td>
<td>0.00873</td>
</tr>
<tr>
<td>producer artist</td>
<td>21120</td>
<td>160816</td>
<td>99110</td>
<td>0.213097</td>
<td>0.36109</td>
<td>0.33419</td>
<td>0.02689</td>
<td>0.28414</td>
</tr>
<tr>
<td>formerTeam team</td>
<td>21466</td>
<td>68742</td>
<td>337707</td>
<td>0.312269</td>
<td>0.34508</td>
<td>0.31623</td>
<td>0.02884</td>
<td>0.23732</td>
</tr>
<tr>
<td>musicalArtist writer</td>
<td>8628</td>
<td>36606</td>
<td>111408</td>
<td>0.235699</td>
<td>0.33847</td>
<td>0.33306</td>
<td>0.00541</td>
<td>0.25869</td>
</tr>
<tr>
<td>nationality birthPlace</td>
<td>9340</td>
<td>44606</td>
<td>432741</td>
<td>0.209389</td>
<td>0.09764</td>
<td>0.10227</td>
<td>−0.00462</td>
<td>0.07720</td>
</tr>
<tr>
<td>writer director</td>
<td>17968</td>
<td>111408</td>
<td>60210</td>
<td>0.298422</td>
<td>0.08033</td>
<td>0.06330</td>
<td>0.01702</td>
<td>0.05636</td>
</tr>
<tr>
<td>developer publisher</td>
<td>4936</td>
<td>21467</td>
<td>36909</td>
<td>0.229934</td>
<td>0.06114</td>
<td>0.06794</td>
<td>−0.00680</td>
<td>0.04708</td>
</tr>
<tr>
<td>birthPlace deathPlace</td>
<td>25485</td>
<td>432741</td>
<td>104257</td>
<td>0.244444</td>
<td>0.05543</td>
<td>0.06178</td>
<td>−0.00634</td>
<td>0.04188</td>
</tr>
</tbody>
</table>

4.4 Pertinence

As described thus far, the distantly supervised training process implicitly made the naive assumption that an instance of a relationship is determined by subject and object alone. This assumption obviously does not always hold. For example, the fact corpus may contain a statement about a person being born in one place and another statement about the person having died in the same place. Since the pattern-finding algorithm is agnostic about the semantics of a pattern, it will extract the same patterns for both facts. An intuitive solution to avoiding errors due to extensional overlap is to leave out duplicate subject-object pairs from the training data. However, it is likely that multiple relationships and their textual representations still exist between many of the remaining concept pairs even when they are not formalized in the training set. A more robust solution is to include duplicates in the training and have measures that can detect both kinds of similarity after the pattern extraction. This is especially important since LoD is seen as streaming, rather than static data and it is assumed that at runtime no informed decision can be made about which facts should be considered for training and which should not. Since LoD only contains positive assertions, the algorithm also cannot rely on negative examples to resolve the ambiguities that inevitably will occur in the extracted patterns. Here, the challenge is to identify and weaken the predictive power of patterns that were incorrectly extracted for a relationship type, because they belong to another relationship that just happened to be extensionally overlapping.

In addition to extensionally similar types of relationships, many others are \textit{intensionally similar}. For example, \textit{physical part of} entails \textit{part of}. Intensional similarity gives a different challenge to a classifier, because here the classifier should not just emphasize the patterns that distinguish the relationships, but also maintain predictive power of patterns that indicate both relationships.

The pertinence measure for relationship patterns was developed to account for semantically similar relationship types. It is inspired by the pertinence measure for term-pairs described in [48]. It boosts the probability of patterns, if they have a high probability of indicating a specific relationship, even though the pattern is shared among different relationship types. Conversely, it diminishes the probability of patterns that are shared because of extensional overlap.

The intuition behind pertinence is that of probability over a semantic space rather than a fixed class. Usually, a probabilistic approach assigns probabilities to each feature (in this case a pattern) in such a way that the probabilities for each class the feature participates in add up to 1. Using Pertinence, these values can add up to more than 1 when the relationship classes it indicates are semantically overlapping.
Pertinence has the effect that similar relationships do not penalize each others shared patterns, whereas dissimilar relationships that share the same patterns get lower scores for these patterns.

The pertinence measure in Equation 17 achieves this adjustment of pattern probabilities. The measure is technically a modified conditional probability computation using a weighted sum in the denominator to take relationship similarities into account.

\[
\overline{p}(R_i|P_j) = \frac{p(P_j|R_i)}{\sum_{k=1}^{m} p(P_j|R_k) \cdot g(1 - \text{sim}_{rel}(R_k, R_i))}
\]  
(17)

The factor \((1 - \text{sim}_{rel}(R_k, R_i))\) grows the more dissimilar a relationships \(R_k\) is to \(R_i\) and thus reduces the impact of \(P_j\) on \(R_i\). The function \(g : [0..1] \rightarrow [0..1]\) can be any monotonous weighting function. It has proven useful to use a logistic function that amplifies closeness and distance of vectors.

Intensional similarity yields a similar textual representation of relationships in the form of patterns [48]. Therefore, the \(\text{sim}_{rel}\) function can compute similarity based on the pattern-probability-vector representation of the relationships. Turner [48] uses a cosine distance between row vectors that represent analogous word pairs. Here, the cosine between rows that represent relationships is computed on the Singular Value Decomposition (SVD) of the \(R^2P\) matrix, because SVD inherently identifies highly descriptive latent dimensions in the data, is known to detect co-importance of features and helps to reduce noise. Since the SVD matrix does not follow the probabilistic framework it is only used in this similarity computation, instead of performing the classification itself using a decomposed matrix. The intensional similarity (\(\text{sim}_{int}\)) computation is shown in Equation 18, where \(\text{cos}_{\text{SVD}}(R_k, R_i)\) is the cosine of the pattern vectors that describe the relationships \(R_k\) and \(R_i\) in the SVD decomposition of \(R^2P\) (described in section 4.3).

\[
\text{sim}_{int}(R_k, R_i) = \text{cos}_{\text{SVD}}(R_k, R_i)
\]  
(18)

The inverse to the premise that intensionally similar relationships share the same patterns does not hold, as purely extensional similarity between two relationships (e.g. the birthPlace and deathPlace relationships) also yields shared patterns. Thus a counterbalance has to be found to the \(\text{sim}_{int}\) measure. Extensional similarity can be computed by dividing the number of shared \((S, O)\) pairs by a function \(f\) of the total number of facts that instantiate the two relationships (Equation 19). The impact of the choice of \(f\) can be compared in Table 5. The “Fraction_{min}” column shows the \(\text{sim}_{ext}\) score for \(f(R_k, R_i) = \min(|R_k|, |R_i|)\). Other options for this function are \(f(R_k, R_i) = \max(|R_k|, |R_i|)\) and \(f(R_k, R_i) = \text{avg}(|R_k|, |R_i|)\) and \(f(R_k, R_i) = \max(|R_k|, |R_i|)\), where \(\hat{R}\) indicates the extension of \(R\).

\[
\text{sim}_{ext}(R_k, R_i) = \frac{|\{r_k(a, b) \in \hat{R}_k \exists r_i(a', b') \in \hat{R}_i, a = a', b = b'\}|}{f(R_k, R_i)}
\]  
(19)

The overall relational similarity is then computed as the intensional similarity \(\text{sim}_{int}\) weighted by 1 minus the extensional similarity \(\text{sim}_{ext}\). See Equation 20. The weighting of the intensional similarity diminishes a false attribution of relational similarity, just because \((S, O)\) pairs are shared.

\[
\text{sim}_{rel}(R_k, R_i) = \text{sim}_{int}(R_k, R_i) \cdot (1 - f_{ext}(\text{sim}_{ext}(R_k, R_i)))
\]  
(20)

The function \(f_{ext} : [0..1] \rightarrow [0..1]\) is used to adjust the importance of the extensional similarity weight. An logistic function is a good choice here, as well. Analogous to the computation of the \(R^2P\) matrix in Equation 16, the pertinence-adjusted matrix \(\overline{R^2P}\) is computed according to Equation 21 using the pertinence computation from Equation 17:

\[
\overline{R^2P}_{ij} = \overline{p}(R_i|P_j)
\]  
(21)

### 4.4.1 Pertinence Analysis

Table 5, besides the above described extensional measures, shows the intensional similarity computed by the cosine distance between relationship vectors. The three rightmost columns in the table show the similarity after applying pertinence (+\(\text{sim}_{int}\)), before applying pertinence (−\(\text{sim}_{int}\)) and the difference in similarity. A positive difference indicates that the pertinence computation yielded a greater similarity than non-pertinence and vice versa.

The top half of Table 5 shows examples of taxonomic relationships from the biology domain. The pertinence computation adjusted pattern probabilities such that the top three relationships pairs became more similar. These three pairs are also immediate ancestors in the taxonomic classification (Species ⇒ Genus ⇒ Family ⇒ Order ⇒ Class ⇒ Phylum ⇒ Kingdom ⇒ Domain), whereas the others are at least one step removed. The lower half of Table 5 gives a good indication of the ameliorating effect of pertinence on relationships that have high extensional overlap. All \(\text{sim}_{rel}\) values are significantly lower than the original −\(\text{sim}_{int}\) values.

Figure 6 shows the impact of pertinence on fact extraction. As anticipated, pertinence leads to higher recall in the lower-confidence regions, because the probability of individual features is increased. However, because of the extensional similarity measure that is applied as part of the pertinence computation, some pattern-probabilities are lower, which leads to higher precision throughout the range of confidence thresholds.
4.5 Relationship Domain and Range probabilities

Often the type of relationship we want to express in natural language is not given by a pattern in a sentence alone, but is also dependent on context. We know that the verb “broke” indicates a different predicate in “Marcus broke the glass” and “Marcus broke the law”. The context here is that the law can be subject to violation and a glass can be shattered. This context is part of our world knowledge, but it can also formally be expressed in ontologies. A classifier needs to know how likely it is that glass or law is an object to the relationship violate or the relationship shattered to guide classification in the right direction.

In well-defined ontologies, domain and range of relationship types are provided in the form of restrictions. A relationship birthplace, for example is defined for the domain Person and the range Location. For LoD data a well-defined ontology may not be present and the possibility of seeing a relationship between instances of two classes must be inferred. Equation 22 shows the conditional probability computation of a relationship R given the types $T_S$ and $T_O$. Triples are shown as $\langle S, R, O \rangle$, with an asterisk indicating a wildcard that fits all possible subjects, objects or relationships. $S \in T_S$ and $O \in T_O$ indicate that $T_S$ and $T_O$ are direct or indirect types of a subject S or object O. G indicates all triples in the training data.

$$p_{\text{domain}}(R|T_S) = \frac{|\{(S, R, \ast) \in G|S \in T_S\}|}{|\{\ast, \ast, \ast \in G|S \in T_S\}|}$$

$$p_{\text{range}}(R|T_O) = \frac{|\{(\ast, R, O) \in G|O \in T_O\}|}{|\{\ast, \ast, O \in G|O \in T_O\}|}$$

Even though an ontology exists for the DBpedia dataset, it is not yet comprehensive enough to account for all entities in DBpedia and it does not provide restrictions for all types of relationships. Moreover, and most importantly, the classes in the ontology do not fully reflect the category hierarchy on Wikipedia that is used for the domain hierarchy creation in Section 3. The domain-range probability computation assures that an estimate for all types of relationships that can be encountered at every category on Wikipedia can be found. The challenge in this computation is the nature of the Wikipedia category graph, which is highly interconnected and does not provide a tree or even lattice structure. Whereas it is tempting to think of the category hierarchy as a class hierarchy, the category links are often rather associative in nature and do not express type or inheritance relationships. For example, Sir Tim Berners-Lee is categorized, amongst others in both “British Computer Scientists”, which is a correct classification, as well as “HTTP”, which is an associative relationship. Taking the latter as a classification makes Tim Berners-Lee a type of Internet protocol and conversely assigns a possibility of an Internet protocol having a birth place or spouse. However, despite this kind of noise, the domain and range probabilities are higher for correct domain and range classes. This is sufficient for the domain and range probabilities because they are merely meant to steer the classification in the right direction when the patterns alone are ambiguous.

In the vector space representation, the values for $p(R|T_S)$ and $p(R|T_O)$ are stored in 2 relationship-prior matrices. The relationship-domain-prior matrix $M_D$ is then of the form $M_{D_{ij}} = \max_{t \in T_S} p(R_j|t)$ and the relationship-range-prior matrix $M_R$ is of the form $M_{R_{ij}} = \max_{t \in T_O} p(R_j|t)$. The $p(R|T_S)$ and $p(R|T_O)$ values can be pre-computed for all domain and range classes over all relationships from the LoD data. The specific $M_D$ and $M_R$ matrices are then filled with these values.

4.6 Matrix-Based Fact Extraction

The extraction of new statements from text is formalized as a classification of concept pairs into relationship types. Given a set of concept pairs, pattern-features for these pairs are extracted from text as described in Section 4.3, step (1). This gives us a concept pair - pattern matrix $CP_2P$ that contains weighted pattern frequencies. In order to have a row sum of 1, the rows in the matrix are normalized according to Equation 23 such that every field in $CP_2P$ contains the probabilities $p(P_j|\langle S, O \rangle)$. Equation 24 then computes the probabilities of each relationship being instantiated by the concept pair using the pertinence computation from Equation 17 for $\tilde{p}(R_j|P_k)$.

$$CP_2P_{ij} = p(P_j|\langle S, O \rangle)$$

$$\tilde{CP}_2P_{ij} = \frac{CP_2P_{ij}}{\sum_{k=1}^{n} CP_2P_{ik}}$$
\[ p(R_j, \langle S, O \rangle_i) = \sum_{k=1}^{m} p(P_k | \langle S, O \rangle_i) \cdot \overline{p}(R_j | P_k) \]  

(24)

In practice, Equation 24 is computed using matrix multiplication (Equation 25). Hence the probabilities of every concept pair instantiating each relationship type is done in one computation step. The resulting concept pair - relationship matrix can thus also be expressed in terms of Equation 24, i.e. $CP2R_{ij} = p(R_j, \langle S, O \rangle_i)$.

$$CP2R = \overline{CP2} \times \overline{R2P^T} \quad \text{(25)}$$

Taking the domain and range probabilities into account, the final matrix $\overline{CP2R}$ is then computed according to Equation 26 by performing a Hadamard (entry wise) multiplication over $CP2R$ and the domain and range prior matrices $M_D$ and $M_R$ (see Section 4.5).

$$\overline{CP2R} = CP2R \circ M_D \circ M_R \quad \text{(26)}$$

The complexity of computing $CP2R$ is $O(nmp)$ with $n$ being the number of concept pairs, $m$ the number of relationships and $p$ the number of patterns. However, since the pattern representations for each concept pair are very sparse, the average complexity is of the order $O(nmc)$ with $c \ll p$. We found that $c$ is on average 11.4 based on the distinct patterns found per concept pair. That the complexity is in practice square rather than cubic makes the algorithm very efficient.

### 4.7 Pattern Analysis

The probabilistic “white-box” approach allows us to analyze the impact of individual patterns on the classification and to compare the patterns that were found highly indicative to patterns found in related work.

In Hearst’s early work on pattern-based hyponym extraction [21], very few hand-picked high-precision patterns were used to extract hyponyms from a text corpus. The recall was reasonably low. Many approaches to pattern-based extraction followed the idea of using a few hand-picked high-quality patterns. Most successful applications of these kinds of patterns are in extracting linguistic relationships.

One goal of this work was to broaden the pattern-base to achieve higher recall in extraction. Analysis of the patterns we found for different types of relationships confirmed the predictive power of many patterns that were identified by Hearst or that we would intuitively attribute to these relationships. For example, “(Subject), such as (Object)” is a good indicator of a sub-class relationship. However, when these patterns are used for taxonomy induction, the ambiguity of this pattern becomes evident. Besides sub-class relationships, it can also indicate an occupation relationship, as in “modern artists, such as Picasso ...”. The algorithm also finds more domain-specific patterns. For example a pattern used in species classification: “(Object) of the family (Subject)”. Other patterns are domain independent and less predictive, but still appropriate, for example “(Object)s, including (Subject)”.

Apart from sub-class/hyponymy-type relationships, we analyzed patterns for other types of relationships. The pattern “(Subject) was born in (Object)”, for example, always indicated the birthplace relationship. However, “(Subject), the former president of (Object)” and “(Subject), prime minister of (Object)” are also good indicators of a birthplace or nationality relationship. Not only do many countries require their presidents or prime ministers to be born in the country, most presidents and prime ministers were actually born in their country. The pertinence measure ensures that the likelihood that these patterns indicate birthplace are less affected by the occurrence of the same patterns in the president or prime minister relationships than a straightforward conditional probability computation would. This broadening of the semantics of a pattern beyond its immediate intension has usually been done using rule-based reasoning on top of the extraction, for example in SOFIE [44]. The problem there is that rules need to be asserted manually and reasoned on separately, whereas in our case these implications are built into the extraction itself.

Other patterns that show a fairly high precision, are of the general form “(name), (location) (profession)”, as the examples of patterns and their significance to the birthplace and occupation relationships in Table 6 show.

As described in [25], doubly anchored patterns, as in “(Subject), american singer and (Object)” perform with very high precision. The general form of these patterns for the occupation relationship is “(name), (location) (profession1) and (profession2)”. As the examples in Table 6 show, the location information can in this case be omitted without harming the indicative power of the pattern.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Pattern</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>birthplace</td>
<td>(\langle \text{Subject}, \langle \text{Object} \rangle ) comedian (\langle \text{Subject}, \langle \text{Object} \rangle ) actor (\langle \text{Subject}, \langle \text{Object} \rangle ) composer (\langle \text{Subject}, \langle \text{Object} \rangle ) guitarist</td>
<td>0.7444144 0.6413878 0.5161507 0.5161426</td>
</tr>
<tr>
<td>occupation</td>
<td>(\langle \text{Subject}, \langle \text{Object} \rangle ) english (\langle \text{Subject}, \langle \text{Object} \rangle ) russian</td>
<td>0.7383892 0.7133882</td>
</tr>
<tr>
<td>occupation,</td>
<td>(\langle \text{Subject}, \langle \text{Object} \rangle ) American singer and (\langle \text{Object} \rangle )</td>
<td>1.0</td>
</tr>
<tr>
<td>doubly</td>
<td>(\langle \text{Subject}, \langle \text{Object} \rangle ) * singer and (\langle \text{Object} \rangle )</td>
<td>1.0</td>
</tr>
<tr>
<td>anchored</td>
<td>(\langle \text{Subject}, \langle \text{Object} \rangle ) * comedian and (\langle \text{Object} \rangle )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Some patterns, on the other hand, are too weak to provide a classification into specific relationships by themselves. Such patterns indicate high-level property types that may not even be in the training corpus. For example, the pattern “(Subject) from (Object)” generally indicates a relationship of spatial or conceptual origin. Such a general pattern has little impact on the precise detection of a relationship type, but can
indicate a general direction for the classifier.

Broadening the pattern base ensures better recall on the one hand and more confidence in the extracted relationships on the other. Hearst patterns are often used for taxonomy induction, but can miss the point, because hypernym/hyponym relationships do not always translate into superclass/subclass relationships. Often we want to instead extract an occupation relationship or a family or genus relationship in biology domains. The combination of many patterns allows a finer-grained classification and the pertinence measure allows multiple classifications with high certainty. This makes it possible to build classifiers for large numbers of relationship types.

4.8 Discussion

In this IE work we explore how surface-pattern-based extraction can be improved by elevating the extraction to a concept level, rather than staying at the term-level. This approach poses the challenge of resolving ambiguous concept identifiers, but also gives the reward of being able to map extracted facts to formal domain models. Technically, the approach allows for efficient processing of textual data, especially when the sources are already indexed and patterns can be pulled from term-position vectors rather than from raw text. The major challenge of the approach from a machine-learning point of view is that we assume no negative training examples and at this stage also no user intervention to correct the algorithm in case of misclassified instances. This is addressed using statistical techniques that detect relationship boundaries and similarities. A drawback when using a pattern-based representation with a maximum-length pattern is that we miss out on information expressed in longer phrases. However, [51] shows that even in a parsing-based systems that uses a parse-tree-generalization algorithm, the extraction accuracy decreases with longer sentences.

5. DOMAIN MODEL EXTRACTION

This section describes the creation of a connected domain model as a combination of Domain Definition (Section 3) and Domain Description (Section 4).

After a domain hierarchy is created based on a keyword description and possibly modified to best reflect the user’s expectations, the individual concepts in the hierarchy (i.e. the leaf nodes or instances) are listed and paired. Recall that this work regards fact extraction as a classification task from concept pairs into relationship types. The pairing of concepts from the hierarchy can be done exhaustively, which will result in high processing cost when searching for patterns that express the concept pairs, because \( n^2 \) concept pairs will have to be considered given that \( n \) concepts are in the model. Thus it is beneficial to restrict the concept pairs that are considered for fact extraction to those that are likely related.

5.1 Concept Pairing heuristic - Wikipedia

For domain hierarchies extracted from Wikipedia, a simple heuristic can be used to select a set \( CP \) of concept pairs in which each pair could be related by one of the trained relationship types. We can assume that if two concepts are directly related, the corresponding Wikipedia pages will be linked. Equation 27 formalizes this idea. There, a concept pair \( (C_i, C_j) \) are concepts from the set of domain concepts \( C \); \( a_{C_i} \) represents a Wikipedia article that describes the concept \( C \) and \( \text{link}(a, b) \) means that a link exists from article \( a \) to article \( b \).

\[
CP = \{C_i, C_j \in C|\exists a_{C_i}, a_{C_j} \in A, \text{link}(a_{C_i}, a_{C_j})\}
\]  

(27)

5.2 Concept Pairing heuristic - General Case

When the concept definition is extracted from a corpus that does not have unambiguous concept links, a pre-selection of the set \( CP \) of concept pairs can be done using co-occurrence analysis on the text corpus that is used. Equation 28 shows a possible pre-selection strategy using pointwise mutual information (PMI). An outcome of \( \text{pmi}(C_i, C_j) > 0 \) indicates that \( C_i \) and \( C_j \) are dependent. The higher the threshold \( \epsilon_{pmi} \) is set, the more dependent \( C_i \) and \( C_j \) will be and hence the more likely they will be related. The joint probability \( p(C_i, C_j) \) that is part of the standard PMI definition can be computed by counting co-occurrence of terms that denote \( C_i \) and \( C_j \) in each document in the corpus.

\[
CP = \{C_i, C_j \in C|(C_i \neq C_j) \land \text{pmi}(C_i, C_j) > \epsilon_{pmi}\}
\]  

(28)

Different from the Wikipedia case, free text only contains concept mentions and does not reference the concepts directly. For this reason, PMI is computed over the co-occurrence of identified concept labels in a set of documents \( D = \{d_1, ..., d_n\} \). Equation 29 shows the PMI computation for this case. \( L_C \) means therein that the label \( L \) is a possible concept denotation for \( C \). Given that only patterns of limited length play a role in the IE algorithm, it is appropriate to only consider occurrences of terms that denote \( C_i \) and \( C_j \) if they are within a small window of tokens, that is, if the distance between the labels that denote \( C_i \) and \( C_j \) in document \( d \) is less than a threshold \( t \).

\[
\text{pmi}(C_i, C_j) = \log \frac{|\{d \in D_{L_C_i} \land \{L_{C_i} \land \text{dist}(L_i, L_j) < t\}|}{|\{d \in D: L_{C_i} \in d\}| \cdot |\{d \in D: L_{C_j} \in d\}|}
\]  

(29)

5.3 Model-Creation

Section 4 described the general algorithm for extracting new facts. To connect a taxonomy or a domain hierarchy with named relationships, the text corpus is searched for occurrences of the concept pairs that were found using Equation
27 or 28 in conjunction with patterns that were learned during training. The $CP2P$ matrix is built from these occurrences of concept pairs with patterns. Creating $\tilde{CP2P}$ by applying Equation 23 and then performing the matrix multiplication from Equation 25 on $CP2P$ and a trained $R2P$ matrix yields the probabilities for relationships between the considered concept pairs.

6. EVALUATION

In this section we evaluate the presented algorithms extensively by analyzing the results of the Information Extraction for Domain Description separately and then evaluate the connected domain models. An evaluation of the Domain Definition extraction can be found in [46].

1. Quantitative evaluation of the facts extracted by the pattern-based method wrt. DBpedia and UMLS gold standards (Section 6.1)

2. Qualitative evaluation of the connected domain models (Section 6.2)

6.1 Fact Extraction Evaluation

This section presents an evaluation of the pattern-based IE algorithm as a whole as well as the pertinence computation in particular. The evaluation is done with respect to two gold standards that are described in section 6.1.1. Any evaluation with respect to gold standards with only positive examples will have the disadvantage of only showing true and false positives. However, it provides an analysis with a significant number of examples.

6.1.1 Datasets

The IE approach taken in this work applies to any combination of corpora, given that there is a training corpus with facts (triples or other representation) that instantiate different relationship types and a text corpus that has textual manifestations of many of the facts, so textual patterns can be learned. To demonstrate the applicability of the algorithm in different domains, the following two combinations of fact and text corpora were chosen:

- DBpedia Infobox fact corpus and Wikipedia text corpus
- UMLS fact corpus and MEDLINE-abstracts text corpus

The contents of both dataset combinations are very different. Wikipedia is an encyclopedia that provides a broad cross-section of human knowledge with currently about 4 Million articles. The DBpedia Infobox corpus [4] contains mostly political, geographical, biographical and entertainment-related facts. MEDLINE is a collection of over 18 million biomedical publications. The abstracts to these publications are freely available and are used in this work. UMLS [27] is a collection of facts from more than 100 different controlled vocabularies and databases in the biomedical field. It covers taxonomic and meronomic relationships as well as relationships describing drug interactions, disease-sites and many more. The scope of UMLS is much narrower, but a lot deeper and more detailed than that of the DBpedia Infobox corpus. The same holds for the corresponding text corpora. Wikipedia contains a broad collection of general knowledge articles whereas MEDLINE contains focused and detailed scientific publications. Neither Wikipedia nor the MEDLINE abstracts were significantly preprocessed with the exception that all Wiki-specific syntax, links, Infoboxes and category assignments were stripped from the text to assure that all pattern representations were taken from raw text rather than from a structured representation.

In both cases, the fact corpus and the text corpus are only loosely connected, insofar as it is known that many of the fact triples are expressed in the form of raw text. Using loosely connected corpora for supervised training has recently been termed Distant Supervision [31]. This loose coupling between corpora brings about advantages and disadvantages. On the one hand it allows us to access unprecedented amounts of training data in the form of facts that were asserted either by a community of experts in the case of UMLS or a broad community of experts and laypersons in the case of DBpedia. These datasets are likely to keep growing and thus provide more and more training facts. On the other hand, since the text is not annotated, it is uncertain that the features that are garnered from the text corpus are representations of the facts in the fact corpus.

Since the objective of fact extraction in this paper is extracting named relationships between entities/concepts in a hierarchy, datatype properties were removed. Moreover, only types of relationships that were instantiated by a minimum of 25 fact triples were taken into consideration. In our opinion, this does not violate the premise that the datasets used should not be pre-processed, because the selection is based on fixed criteria, not on the quality of the facts or their availability in the corresponding text corpus.

6.1.2 Gold Standard Evaluation

For this evaluation all patterns for all distinct Subject-Object pairs were extracted in both corpora (DBpedia Infobox and UMLS). The resulting matrices were randomly split into 80% training examples and 20% testing examples. Training and testing sets were completely disjoint, i.e. no subject or object that appeared in one set was allowed to be in the other. The splitting was repeated 10 times and the results averaged. The size of a testing set was about 11,000 facts in the DBpedia case and about 4100 in the UMLS case. Precision and recall were first computed on a per-relationship basis and then averaged over all relationships. This normalized evaluation yields lower values, because relationships with many examples tend to perform with higher precision and recall. Due to the normalization, these types of relationships are counted equally to relationship types with very few examples, thus...
Figure 7: Precision and recall depending on a confidence threshold $\epsilon_{rel}$: DBpedia test set and Wikipedia text corpus.

Figure 8: Precision and recall depending on a confidence threshold $\epsilon_{rel}$: UMLS test set and MEDLINE text corpus.

diminishing the impact of each correctly extracted fact from an abundantly represented relationship. However, for the evaluation of an open IE approach this is more meaningful than averaging over all extracted facts, because it shows that even higher numbers of relationship types do not break the system.

Figures 7 and 8 show precision and recall with respect to the gold-standards DBpedia and UMLS. The horizontal axis indicates the confidence threshold $\epsilon_{rel}$ that was used, i.e. only statements that were extracted with a probability greater than $\epsilon_{rel}$ were taken into consideration. The average values show the arithmetic mean precision and recall values over all relationship types, the max values show the maximum precision and recall among the relationship types. Precision and recall are thereby computed according to Equations 30 and 31.

$$\text{Precision}_{\epsilon_{rel}} = \frac{1}{|R|} \sum_{R \in R} \frac{|\text{correct facts for } R, \text{ confidence } > \epsilon_{rel}|}{|\text{all extracted facts for } R, \text{ confidence } > \epsilon_{rel}|}$$ (30)

$$\text{Recall}_{\epsilon_{rel}} = \frac{1}{|R|} \sum_{R \in R} \frac{|\text{correct facts for } R, \text{ confidence } > \epsilon_{rel}|}{|\text{all gold standard facts for } R|}$$ (31)

Figure 7 shows the automatic evaluation of precision and recall over all cross-evaluation sets of the DBpedia-Wikipedia corpus. Considered were only those relationship types for which more than 25 possible occurrences were found in the Wikipedia corpus, which amounted to an average of 107 distinct types. Only direct hits in first rank according to Equation 32 were taken into account.

$$R_{CP_i} = \arg \max_j p(R_j | CP_i) \text{ if } p(R_j | CP_i) > \epsilon_{rel}$$ (32)

The evaluation shown in Figure 8 over the UMLS-MEDLINE corpus is analogous. The curves are in general steeper than in the DBpedia-Wikipedia case and go up to over 80% precision as the confidence threshold increases. A pattern analysis showed that in a scientific corpus the expressions are more specific. This translates to more specific patterns that apply to fewer concept pairs. The precision and recall lines cross at comparable points in both cases, which indicates that there is a baseline of patterns that describe more general types of relationships. In all cases the results were well over the random baseline. Even in the high recall regions the average precision is at least 35% and goes up to 65% as the confidence threshold increases, with some relationship types showing perfect precision. The high recall especially in the DBpedia case is a definite improvement over the approaches mentioned in the related work section that are more precision-oriented. It can thus be shown that with a basic probabilistic approach the surface pattern analysis can be used to augment domain models with new facts for information retrieval applications.

To show the difference in performance when changing the evaluation style, in the following evaluation the precision was averaged over all extracted facts without first averaging over the relationship types. Figure 9 shows these results. It is apparent that the precision increased significantly, especially in the DBpedia case. Using an oversampling strategy for underrepresented types of relationships rather than random sampling would remedy this discrepancy for evaluation purposes. However, when looking at LoD as constantly streaming data, rather than a static corpus, it is more realistic not to make assumptions about the distribution of relationship
types, because the distribution changes depending on editing dynamics in the community. In an application scenario the classifier would naturally be trained on all available facts to have a maximum gain of pattern probability information at each point in time and thus it is bound to the distribution of relationships types in the dataset that it is dealt.

6.1.3 Comparison to other approaches
As outlined in the related work section, the information extraction approach taken here makes different assumptions concerning the training data and uses different strategies in processing the data and in training classifiers than its competitors. It is thus difficult to make a fair comparison. However, the work by Mintz et al. [31] is a good comparison insofar as it uses a similar training strategy on comparable data. The comparison (see Figure 10) was done by averaging across all 107 DBpedia relationships (Mintz uses 102). For this comparison, a more precision-oriented and a more recall-oriented experiment were performed. In the precision-oriented run the patterns were not generalized, whereas in the recall-oriented run a generalization was performed with up to two wild cards per pattern. The precision-oriented experiment peaks at a higher precision than Mintz, but has a sharper drop until it maintains a higher precision in the high recall regions as well. The recall-oriented run has a much smoother curve, but starting at a lower precision than Mintz. It however maintains a higher precision starting at a recall of about 0.2. Mintz et al. use a “multi-class logistic classifier optimized using L-BFGS with Gaussian regularization”. The results suggest that, even with a weaker pattern representation, our classifier using conditional probabilities modified by the pertinence measure, can outperform this well-regarded ML algorithm.

6.2 Evaluation of the full Domain Models
The full model creation is evaluated in two different application scenarios. The first one is geared towards rapid model creation for general purpose information filtering and browsing tasks, the second one is aimed at guided scientific information retrieval.

6.2.1 Models for Information Filtering and Browsing
To properly evaluate the full model creation for information filtering, we generated six models of different domains. To get a broad spectrum of models, we build them according to the semantic scope dimensions that were identified in [35]:

1. Global vs. Local
2. Compact vs. Loose
3. Deterministic vs. Unexpected
4. Transient vs. Lasting

The models were created for the criteria shown in Table 7. The Query column shows the queries that were sent to the Doozer hierarchy creation. The models were then manually evaluated. The number of concepts that were contained in each model gives an idea of coverage. The consistently high precision for the domain definition shows that the concepts contained in the models were largely relevant to the Domain.
the relationship type was very close to a correct type. For example the triple (Helmut Kohl \rightarrow commander \rightarrow German Chancellery) was counted as positive, even though the chancellor is not a military commander of the chancellery. The recall values are taken relative to the number of correct facts extracted using the lowest confidence threshold. The actual recall as it pertains to the domain is difficult to estimate, because the number of possible facts in a domain is unbounded. Figures 11 and 12 show precision and relative recall of the fact extraction part for these models.

Figure 1 (in the Introduction) showed an excerpt of the model created for the Relationships between India and Pakistan. As in all the other example models, none of the asserted facts was directly taken from DBpedia. By placing the term “Kashmir” into the query, Doozer++ put emphasis on that part of the India-Pakistan relations. The algorithm correctly identified Pakistani leaders. However, it does not distinguish between current leaders and former leaders. This is due to the fact that training set from the DBpedia corpus also does not make this distinction. The extractor found many triples that give important contextual information, such as the fact that the second president of Pakistan, Ayub Khan, was born in (the former British) India or that Pakistan was founded by Muhammad Ali Jinnah. It also correctly identified the region of Kashmir as belonging to both countries, but placed the state of Azad Kashmir into Pakistan and Jammu-and-Kashmir into India.

6.2.2 Evaluation of the Cognitive Science Model
As mentioned in the introduction, we developed a larger domain model for the research area of Human Cognitive Performance. This model currently serves as background knowledge for a domain-centric semantic browser named Scooner [8]. Here, the objective was to have a comprehensive account of the domain with high precision and recall. For this reason, a domain expert provided us with 3 lists of terms that are important in the domain. These lists were from the domains of cognitive psychology and neuroscience, whereby the neuroscience terms were again split into conceptual and functional aspects. An excerpt of these lists can be found in Table 8. The seed query for the model creation was a conjunction of all lists, whereby the terms in each list were disjunct. The query thus looked as follows: (abstract reasoning \ OR abstraction \ OR ...) AND (“activated cortical volume” \ OR “activation of frontal cortex” \ OR ...) AND (“acute and chronic hypoxia” \ OR “adult neurogenesis” \ OR ...).

Since the goal of the corresponding project was to have guided browsing of MEDLINE articles using entities and relationships relevant to the domain of human cognitive performance, the embodiment of the hierarchy with facts that instantiate domain-relevant relationships was critical. Here, precision was more important than in the previous examples, even though, as a browsing guideline, recall was still the primary focus. To connect the entities in the hierarchy with named relationships from the biomedical domain, the relationship model in the form of the $R^2P$ matrix was trained by extracting patterns from the MEDLINE abstracts text corpus based on facts available in the UMLS corpus. Figure 2 in the introduction showed a small excerpt of the connected model. Figure 14 shows a more strongly connected part of the extracted concept graph. Of particular interest for the cognitive science domain are the receives_input_from and sends_output_to relationships that indicate interaction of brain regions in this model. These relationships do not exist at all in DBpedia and whereas some of them can be found in UMLS, the UMLS concepts cannot always be mapped to the coarser Wikipedia concepts. Figure 13 shows the first five levels of the resulting category hierarchy in a Treemap representation. Spatial inclusion in the Treemap represents a subcategory relationship, the size of the rectangles indicates the number of descendants and hence the importance of the category to the domain.

We examined the extracted facts and removed any that were already available in the UMLS training corpus from the sample that was evaluated. Cognitive scientists and biologists at the Air Force Research Lab evaluated 415 extracted facts scored each fact on a scale of 1 to 9; 1 being plain wrong
and 9 being true, novel and interesting. Figure 15 shows the scoring. It displays the percentage for each score and cumulative percentages for scores 1-2 (incorrect: 21%) and 3-9 (correct: 79%) respectively. About 30% of the extracted facts was deemed novel and interesting. The scoring rationale is as follows:

7-9: Correct Information not commonly known
5-6: General Information that is correct
3-4: Information that is correct, but trivial
1-2: Information that is overall incorrect

6.3 Discussion
Analysis of the fact extraction shows that a high percentage of the facts deemed as novel and interesting were extracted based on highly specialized low frequency patterns. These patterns appear in the text corpus few times with the provided training facts and tend to appear with subject and object concepts that fall in the same domain classes or range classes respectively. Figure 16 shows the average score for extracted facts for each confidence score from 1 to 9. The figure shows that high-quality facts get highest scores, but also some incorrect facts were extracted with high confidence. This is because within the pattern distribution, those patterns that tend to identify particular relationships with high confidence are in the long tail of the pattern frequency distribution. However, noisy patterns also tend to occupy this space. With the absence of negative examples, these are not easily distinguished. An analysis of the facts that were incorrectly identified with high confidence shows that they largely fall in two categories. The first is that of a formally incorrect but metaphorically correct relationship or of generally very high relatedness. For example, the extracted assertion (Interpeduncular_Cistern → disease_has_associated_anatomic_site → Cerebral_peduncle) is incorrect, because...
the Interpeduncular Cistern is not a disease. However, it does have the associated anatomic site Cerebral peduncle.

The second is that of incorrect directionality. In many cases the asserted relationship is correct, but points in the wrong direction. For example, the assertion \( \text{Pituitary Gland} \rightarrow \text{sends output to} \rightarrow \text{Supraoptic nucleus} \) is incorrect because the supraoptic nucleus sends output to the pituitary gland, not vice versa. Often, when these directional relationships are described in text, the direction is expressed in the context rather than in the phrase that relates the two entities. A possible remedy for these kinds of errors that are caused by patterns incorrectly identified as high-quality even though they are either noise or belong to a different type of relationship is to retrain the classifier with these incorrectly classified statements as negative examples. The NELL project [10] uses this kind of active learning to improve its precision. For future work we will incorporate an active learning component, as well.

An interesting aspect of the analysis of incorrectly identified relationships in sections 6.2.1 and 6.2.2 is that many of them represent the correct relationship qualia [13]. A business merger is often referred to similarly to a marriage, holding political office is often described similarly to holding a military rank or being a military commander. Thus it makes sense to connect the merging companies with the spouse relationship or the chancellor to the chancellery with the commander relationship. Lakoff and Johnson [26], amongst others, identified evidence that the use of basic metaphors in natural language is ubiquitous. Turney’s work on word pair analogies[48] also provides statistical significance to these claims. When encountering analogous relationships such as the ones mentioned, more well-formalized background knowledge will allow us to rule out wrong types of relationships, when we know that e.g. the domain and range of the spouse relationship must be Person.

7. RELATED WORK

IE from a structured corpus: Research efforts that have made use of the Wikipedia corpus to infer taxonomic knowledge, such as [34], often use Hearst-style patterns and heuristics based on Wikipedia naming conventions to identify those inter-category relationships in the Wikipedia hierarchy that are actually is a relationships and are helpful in distinguishing between classes and instances. Just like our work, these efforts combine top-down and bottom-up extraction, but with the goal of refining an existing corpus, rather than finding knowledge that was previously not formalized. DBpedia [4] and Freebase [5] also make use of information that is available in a structured form on Wikipedia. These projects go beyond the taxonomic structure of Wikipedia and aim at making factual information from the Wikipedia InfoBox data accessible in the form of RDF statements or other structured representations. To some extent, DBpedia also maps synonymous relationships from the InfoBox data to semantically normalized relationships that are formalized in the DBpedia ontology. Another example is YAGO [43] that improves the classification of Wikipedia concepts by mapping them to WordNet [15] and by applying various categorization and disambiguation heuristics. For an application that aims at producing formally correct ontologies, these and comparable efforts constitute important preprocessing steps. However, none of those works is concerned with the restriction of the Wikipedia corpus to a specific domain of interest.

Domain Taxonomy extraction from unstructured text: In the Taxaminer project [24] taxonomies were extracted from biomedical documents with no structural knowledge of the domain available to the system. The resulting hierarchy was generated solely by identifying cohesive clusters in a hierarchy that was an artifact of a bisecting k-Means clustering process. Even though the clusters were of high quality based on IR measures, the resulting hierarchy did not reflect the shared human conceptualization of the domain when it was compared to MeSH. The intuitive explanation for this is that research papers are problem-centric, i.e. the view that is taken of the domain emanates from the problem at hand. A taxonomy is world-view dependent, i.e. it tries to summarize the entirety of concepts that are pertinent to a field of interest.

Taxonomy induction based on seed terms using Hearst-style patterns was done by Sanchez and Moreno in [38] as well as Navigli et al. [32]. Domain taxonomies were extracted by finding instances of these patterns in large numbers of web pages. The created taxonomies are of high quality, which can be attributed to the high-precision nature of Hearst-patterns. However, the amount of pages that has to be crawled and the resulting extraction time is prohibitive for on-the-fly creation of taxonomies. The precision for a scientific domain in [32] was 81.5% and between 75% and 95% in [38], depending on the domain. We naturally achieve a higher precision, because taxonomic relationships are already available in the corpus for the Domain Definition. However, [32, 38] are promising contributions that could replace or augment the current Domain Definition step in order to create deeper and more finely grained domain taxonomies.

The clustering-based approach to taxonomy extraction shows that the way concepts descriptors are distributed across doc-
Acknowledging the importance of a proper classification of concepts and the difficulties involved in bottom-up extraction of taxonomies spawned the decision to extract taxonomies top-down, from existing conceptual sources.

Relationship Extraction from unstructured text: Most bottom-up relationship extraction work uses IE techniques. Figure 17 puts our work in context with the current state of the art in IE. The dimension NLP complexity refers to the amount of text processing that was performed, e.g., POS tagging, parsing, etc. Openness of extraction refers to the number of relationship types that can be extracted, ranging from very few dedicated types, such as hypernym/hyponym relationships to an unbounded number of relationship types. The work described in this paper (Doozer++) is placed in the no-NLP/many relationships corner. Most works have either been restricted in the number of relationship classes that are extracted [1, 12, 21, 33, 41] and/or have made use of parsing or POS-tagging to improve precision and recall of the methods [2, 10, 31, 36, 42, 50, 51].

Pattern-based IE has been successfully applied in past research. Hearst [21] used manually identified patterns that indicate hyponym relationships. Snow et al. [41] use automatically identified pattern vectors for the same task. The Snowball system [1] uses a pattern-definition similar to ours to identify a restricted set of relations and assumes a previous named entity recognition to restrict the extraction to (Organization, Location) pairs. Paşca et al. [33] also heavily restrict the types of relationships by extracting only date-of-birth attributes from Web documents, thus achieving a very high average precision of 93.17%. Whether the system has the potential to scale up to an arbitrary number of property types is not discussed. The pattern-based approach described in this paper is inspired by P.D. Turney’s work on identifying analogous word pairs [48]. Similar to our work, Turney uses vector space representations of surface patterns without parsing or POS-tagging the text.

Structurally open IE approaches that are not restricted to a fixed number of relations are TextRunner [2] and Ramakrishnan’s work on extracting relationships in the BioMedical domain [36]. These systems assume that the syntactic predicate in the sentence expresses the relationship of interest, which is not always the case. Systems that use a relational targeting IE approach in conjunction with NLP techniques are Kylin [50] and WOE [51] which use Wikipedia InfoBoxes to train Conditional Random Fields (CRFs). WOE outperforms TextRunner when using dependency parses for extracting new relationships. Relational-targeting IE usually need labeled corpora with positive and negative training examples, which tends to increase performance significantly. However, labeled data is scarce and often outdated. The approach taken here lines up with Mintz et al. [31], who coined the term “distant supervision” for learning tasks that use relational data and unstructured text. Contrary to ours, though, their IE approach uses both lexical and syntactic features, where a combination of both performs best, especially in higher-recall scenarios. However, even the lexical features go beyond mere surface patterns and contain POS information. Moreover, a named entity tagger identifies a limited number of entity tags, such as Person, Organization and Location. We believe that neither POS tags nor syntactic features will be available at runtime. LEILA [42] performs a strong linguistic analysis of the corpus and identifies complex patterns along paths through the parse tree. LEILA can also take advantage of Wikipedia-specific features, such as common page structuring, category-assignment, titles, headings, links and InfoBoxes. In this work we want to keep the relationship extraction independent of a particularly structured corpus for two reasons. First we extract domain-specific relationships from any specialized corpus and second, we expand the base for the extraction across corpora, i.e., potentially search for evidence on the whole Web.

NELL [9] is a so-called never ending learning system that extracts new relationships by continuously reading the Web. It uses multiple complementary approaches to achieve better precision by coupling different learning functions. It is part of the Read-the-Web project [10], which also uses background knowledge in the form of ontologies or domain models to improve the classification of entities and relationships. Different from our work, it tries to learn new concepts using NER techniques, which leads to some incorrectly identified concepts. However, an active learning component assures the constant improvement of the system.

8. CONCLUSION
This paper presented automatic creation of focused domain models using a 2-step process of hierarchy creation and fact
extraction. Based on the conceptual and technical split of the model creation into Domain Definition and Domain description we created a modular framework that allows us to use different Information Extraction methods for different types of corpora, thus allowing the use of (semi-)formal information where available and extraction of additional information from informal text where necessary. Whereas the first step explicitly takes advantage of a linked and categorized corpus, fact extraction is set up to work on any text corpus. The results show a precision of 79% for named relationships in a previously unknown scientific field of interest and medium to high precisions in non-scientific fields of inquiry. These results in previously unseen domains validated the results obtained by evaluating with respect to UMLS and DBpedia gold standards. This shows that domain model creation can be done successfully in open domains if some error tolerance is given. Most IR applications do not require 100% precision and are thus ideal venues for domain models that are used for browsing or classification.

This paper showed that efficient fact extraction is possible with a framework that does not employ expensive NLP techniques and can thus work on indices rather than full text. The quality of the extraction is in line with state-of-the-art open IE approaches. A distantly supervised, positive-only classifier makes the use of LoD training data possible without human intervention. Facing the problem of missing negative training data, we developed a pertinence measure to ensure that patterns a) are sufficiently discriminative when they are highly indicative of a particular relationship type and b) do not compete when they express relationships with similar semantics.

The precision of the algorithm is comparable to results in related research that were achieved using more involved NLP methods to pre-process the features used for classification rather than use unprocessed surface patterns as in our case. The low-recall problem that most pattern-based IE methods suffer from has been addressed with a wildcard-based generalization technique that improves recall significantly without major loss of precision, even though recall still cannot fully measure up to NLP-based systems. We see one reason for the good precision of this algorithm in the concept-centered approach. That is, starting with an extraction that is restricted to known concepts that have a known textual representation anchors the patterns in a much better way than e.g. a pair of bottom-up identified nouns or noun phrases.

The automatically created model of the cognitive science domain is in active use in a topic-based information retrieval and focused browsing application called Scooner [8]. We are collecting data on users’ browsing behavior to assess whether conclusions about correctly vs. incorrectly extracted can be drawn from the way these facts are used in browsing.

In future research we will incorporate the generated models in content delivery systems, such as Twarql [30], where the models are used to filter desired information. In turn, the streaming information updates the models to account for new developments. Thus, this work provides an important part in the development of a continuous cycle of semantic representations as described in [40]. To improve model creation accuracy, we are planning on incorporating user feedback in an active-learning system as demonstrated for example in [9], so incorrectly extracted facts can be used for retraining with negative training examples. In addition to the explicit feedback, these negative examples will be gained by manual analysis of extracted samples as well as based on assumptions about correctness gained from the browsing behavior on Scooner.

Acknowledgments
This work is partly funded in part by NSF awards IIS-1111182 and a grant by the Air Force Research Lab.

9. REFERENCES


H. Purohit, Y. Ruan, A. Joshi, S. Parthasarathy, and A. P. Sheth. Understanding User-Community
Engagement by Multi-faceted Features: A Case Study on Twitter. *First international workshop on social media engagement (SoME 2011)*, 2011.


