

# LEILA: Learning to Extract Information by Linguistic Analysis

**Fabian M. Suchanek**  
Max-Planck-Institute  
for Computer Science  
Saarbrücken/Germany  
suchanek@mpi.mpg.de

**Georgiana Ifrim**  
Max-Planck-Institute  
for Computer Science  
Saarbrücken/Germany  
ifrim@mpi.mpg.de

**Gerhard Weikum**  
Max-Planck-Institute  
for Computer Science  
Saarbrücken/Germany  
weikum@mpi.mpg.de

## Abstract

One of the challenging tasks in the context of the Semantic Web is to automatically extract instances of binary relations from Web documents – for example all pairs of a person and the corresponding birthdate. In this paper, we present LEILA, a system that can extract instances of arbitrary given binary relations from natural language Web documents – without human interaction. Different from previous approaches, LEILA uses a deep syntactic analysis. This results in consistent improvements over comparable systems (such as e.g. Snowball or TextToOnto).

## 1 Introduction

### 1.1 Motivation

Search engines, question answering systems and classification systems alike can greatly profit from formalized world knowledge. Unfortunately, manually compiled collections of world knowledge (such as e.g. WordNet (Fellbaum, 1998)) often suffer from low coverage, high assembling costs and fast aging. In contrast, the World Wide Web provides an enormous source of knowledge, assembled by millions of people, updated constantly and available for free. Since the Web data consists mostly of natural language documents, a first step toward exploiting this data would be to extract instances of given target relations. For example, one might be interested in extracting all pairs of a person and her birthdate (the `birthdate`-relation), pairs of a company and the city of its headquarters (the `headquarters`-relation) or pairs of an entity and the concept it belongs to (the `instanceOf`-relation). The task is, given a set of Web documents and given a target relation, extracting pairs of entities that are in the target relation. In this paper, we propose a novel method for this task, which works on natural language Web documents and does not require human interac-

tion. Different from previous approaches, our approach involves a deep linguistic analysis, which helps it to achieve a superior performance.

### 1.2 Related Work

There are numerous Information Extraction (IE) approaches, which differ in various features:

- **Arity of the target relation:** Some systems are designed to extract unary relations, i.e. sets of entities (Finn and Kushmerick, 2004; Califf and Mooney, 1997). In this paper we focus on the more general binary relations.
- **Type of the target relation:** Some systems are restricted to learning a single relation, mostly the `instanceOf`-relation (Cimiano and Völker, 2005b; Buitelaar et al., 2004). In this paper, we are interested in extracting arbitrary relations (including `instanceOf`). Other systems are designed to discover new binary relations (Maedche and Staab, 2000). However, in our scenario, the target relation is given in advance.
- **Human interaction:** There are systems that require human intervention during the IE process (Riloff, 1996). Our work aims at a completely automated system.
- **Type of corpora:** There exist systems that can extract information efficiently from formatted data, such as HTML-tables or structured text (Graupmann, 2004; Freitag and Kushmerick, 2000). However, since a large part of the Web consists of natural language text, we consider in this paper only systems that accept also unstructured corpora.
- **Initialization:** As initial input, some systems require a hand-tagged corpus (J. Iria, 2005; Soderland et al., 1995), other systems require text patterns (Yangarber et al., 2000) or templates (Xu and Krieger, 2003) and again others require seed tuples (Agichtein and Gravano, 2000; Ruiz-Casado et al., 2005; Mann and Yarowsky, 2005) or tables of target concepts (Cimiano and Völker, 2005a). Since hand-

labeled data and manual text patterns require huge human effort, we consider only systems that use seed pairs or tables of concepts.

Furthermore, there exist systems that use the whole Web as a corpus (Etzioni et al., 2004) or that validate their output by the Web (Cimiano et al., 2005). In order to study different extraction techniques in a controlled environment, however, we restrict ourselves to systems that work on a closed corpus for this paper.

One school of **extraction techniques** concentrates on detecting the boundary of interesting entities in the text, (Califf and Mooney, 1997; Finn and Kushmerick, 2004; Yangarber et al., 2002). This usually goes along with the restriction to unary target relations. Other approaches make use of the context in which an entity appears (Cimiano and Völker, 2005a; Buitelaar and Ramaka, 2005). This school is mostly restricted to the `instanceOf`-relation. The only group that can learn arbitrary binary relations is the group of pattern matching systems (Etzioni et al., 2004; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002; Brin, 1999; Soderland, 1999; Xu et al., 2002; Ruiz-Casado et al., 2005; Mann and Yarowsky, 2005). Surprisingly, none of these systems uses a deep linguistic analysis of the corpus. Consequently, most of them are extremely volatile to small variations in the patterns. For example, the simple subordinate clause in the following example (taken from (Ravichandran and Hovy, 2002)) can already prevent a surface pattern matcher from discovering a relation between "London" and the "river Thames": "London, which has one of the busiest airports in the world, lies on the banks of the river Thames."

### 1.3 Contribution

This paper presents LEILA (Learning to Extract Information by Linguistic Analysis), a system that can extract instances of an arbitrary given binary relation from natural language Web documents without human intervention. LEILA uses a deep analysis for natural-language sentences as well as other advanced NLP methods like anaphora resolution, and combines them with machine learning techniques for robust and high-yield information extraction. Our experimental studies on a variety of corpora demonstrate that LEILA achieves very good results in terms of precision and recall and outperforms the prior state-of-the-art methods.

### 1.4 Link Grammars

There exist different approaches for parsing natural language sentences. They range from sim-

ple part-of-speech tagging to context-free grammars and more advanced techniques such as Lexical Functional Grammars, Head-Driven Phrase Structure Grammars or stochastic approaches. For our implementation, we chose the Link Grammar Parser (Sleator and Temperley, 1993). It is based on a context-free grammar and hence it is simpler to handle than the advanced parsing techniques. At the same time, it provides a much deeper semantic structure than the standard context-free parsers. Figure 1 shows a simplified example of a linguistic structure produced by the link parser (a *linkage*).

A linkage is a connected planar undirected graph, the nodes of which are the words of the sentence. The edges are called *links*. They are labeled with *connectors*. For example, the connector **subj** in Figure 1 marks the link between the subject and the verb of the sentence. The linkage must fulfill certain linguistic constraints, which are given by a *link grammar*. The link grammar specifies which word may be linked by which connector to preceding and following words. Furthermore, the parser assigns part-of-speech tags, i.e. symbols identifying the grammatical function of a word in a sentence. In the example in Figure 1, the letter "n" following the word "composers" identifies "composers" as a noun.

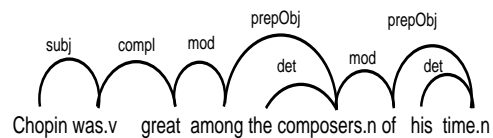


Figure 1: A simple linkage

Figure 2 shows how the Link Parser copes with a more complex example. The relationship between the subject "London" and the verb "lies" is not disrupted by the subordinate clause:

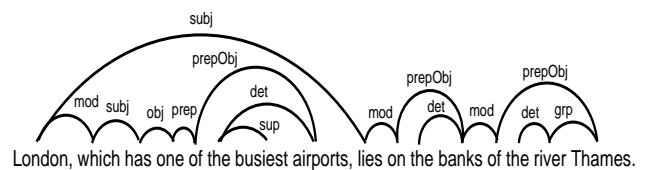


Figure 2: A complex linkage

We say that a linkage *expresses* a relation *r*, if the underlying sentence implies that a pair of entities is in *r*. Note that the deep grammatical analysis of the sentence would allow us to define the meaning of the sentence in a theoretically well-founded way (Montague, 1974). For this paper, however, we limit ourselves to an intuitive understanding of the notion of meaning.

We define a *pattern* as a linkage in which two

words have been replaced by placeholders. Figure 3 shows a pattern derived from the linkage in Figure 1 by replacing "Chopin" and "composers" by the placeholders "X" and "Y".

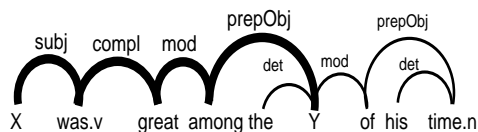


Figure 3: A pattern

We call the (unique) shortest path from one placeholder to the other the *bridge*, marked in bold in the figure. The bridge does not include the placeholders. Two bridges are regarded as equivalent, if they have the same sequence of nodes and edges, although nouns and adjectives are allowed to differ. For example, the bridge in Figure 3 and the bridge in Figure 4 (in bold) are regarded as equivalent, because they are identical except for a substitution of "great" by "mediocre". A pattern *matches* a linkage, if an equivalent bridge occurs in the linkage. For example, the pattern in Figure 3 matches the linkage in Figure 4.

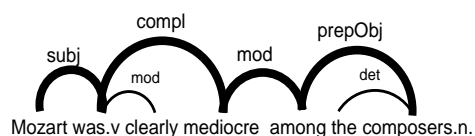


Figure 4: A matching linkage

If a pattern matches a linkage, we say that the pattern *produces* the pair of words that the linkage contains in the position of the placeholders. In Figure 4, the pair "Mozart" / "composers" is produced by the pattern in Figure 3.

## 2 System Description

### 2.1 Document Pre-Processing

LEILA accepts HTML documents as input. To allow the system to handle date and number expressions, we normalize these constructions by regular expression matching in combination with a set of functions. For example, the expression "November 23rd to 24th 1998" becomes "1998-11-23 to 1998-11-24" and the expression "0.8107 acre-feet" becomes "1000 cubic-meters". Then, we split the original HTML-document into two files: The first file contains the proper sentences with the HTML-tags removed. The second file contains the non-grammatical parts, such as lists, expressions using parentheses and other constructions that cannot be handled by the Link Parser. For example, the character sequence "Chopin (born 1810) was a great composer" is split into the sentence "Chopin

was a great composer" and the non-grammatical information "Chopin (born 1810)". The grammatical file is parsed by the Link Parser.

The parsing allows for a restricted named entity recognition, because the parser links noun groups like "United States of America" by designated connectors. Furthermore, the parsing allows us to do anaphora resolution. We use a conservative approach, which simply replaces a third person pronoun by the subject of the preceding sentence. For our goal, it is essential to normalize nouns to their singular form. This task is non-trivial, because there are numerous words with irregular plural forms and there exist even word forms that can be either the singular form of one word or the plural form of another. By collecting these exceptions systematically from WordNet, we were able to stem most of them correctly with our Plural-to-Singular Stemmer (*PlingStemmer*<sup>1</sup>). For the non-grammatical files, we provide a pseudo-parsing, which links each two adjacent items by an artificial connector. As a result, the uniform output of the preprocessing is a sequence of linkages, which constitutes the input for the core algorithm.

### 2.2 Core Algorithm

As a definition of the target relation, our algorithm requires a function (given by a Java method) that decides into which of the following categories a pair of words falls:

- The pair can be an **example** for the target relation. For instance, for the *birthdate*-relation, the examples can be given by a list of persons with their birth dates.
- The pair can be a **counterexample**. For the *birthdate*-relation, the counterexamples can be deduced from the examples (e.g. if "Chopin" / "1810" is an example, then "Chopin" / "2000" must be a counterexample).
- The pair can be a **candidate**. For *birthdate*, the candidates would be all pairs of a proper name and a date that are not an example or a counterexample.
- The pair can be none of the above.

The core algorithm proceeds in three phases:

1. In the *Discovery Phase*, it seeks linkages in which an example pair appears. It replaces the two words by placeholders, thus producing a pattern. These patterns are collected as *positive patterns*. Then, the algorithm runs through the sentences again and finds all linkages that match

<sup>1</sup>available at <http://www.mpii.mpg.de/~suchanek>

a positive pattern, but produce a counterexample. The corresponding patterns are collected as *negative patterns*<sup>2</sup>.

2. In the *Training Phase*, statistical learning is applied to learn the concept of positive patterns. The result of this process is a classifier for patterns.
3. In the *Testing Phase*, the algorithm considers again all sentences in the corpus. For each linkage, it generates all possible patterns by replacing two words by placeholders. If the two words form a candidate and the pattern is classified as positive, the produced pair is proposed as a new element of the target relation (an *output pair*).

In principle, the core algorithm does not depend on a specific grammar or a specific parser. It can work on any type of grammatical structures, as long as some kind of pattern can be defined on them. It is also possible to run the Discovery Phase and the Testing Phase on different corpora.

### 2.3 Learning Model

The central task of the Discovery Phase is determining patterns that express the target relation. These patterns are generalized in the Training Phase. In the Testing Phase, the patterns are used to produce the output pairs. Since the linguistic meaning of the patterns is not apparent to the system, the Discovery Phase relies on the following hypothesis: Whenever an example pair appears in a sentence, the linkage and the corresponding pattern express the target relation. This hypothesis may fail if a sentence contains an example pair merely by chance, i.e. without expressing the target relation. Analogously, a pattern that does express the target relation may occasionally produce counterexamples. We call these patterns *false samples*. Virtually any learning algorithm can deal with a limited number of false samples.

To show that our approach does not depend on a specific learning algorithm, we implemented two classifiers for LEILA: One is an adaptive k-Nearest-Neighbor-classifier (kNN) and the other one uses a Support Vector Machine (SVM). These classifiers, the feature selection and the statistical model are explained in detail in (Suchanek et al., 2006). Here, we just note that the classifiers yield a real valued label for a test pattern. This value can be interpreted as the confidence of the classification. Thus, it is possible to rank the output pairs of LEILA by their confidence.

---

<sup>2</sup>Note that different patterns can match the same linkage.

## 3 Experiments

### 3.1 Setup

We ran LEILA on different corpora with increasing heterogeneity:

- **Wikicomposers:** The set of all Wikipedia articles about composers (872 HTML documents). We use it to see how LEILA performs on a document collection with a strong structural and thematic homogeneity.
- **Wikigeography:** The set of all Wikipedia pages about the geography of countries (313 HTML documents).
- **Wikigeneral:** A set of random Wikipedia articles (78141 HTML documents). We chose it to assess LEILA's performance on structurally homogenous, but thematically random documents.
- **Googlecomposers:** This set contains one document for each baroque, classical, and romantic composer in Wikipedia's list of composers, as delivered by a Google "I'm feeling lucky" search for the composer's name (492 HTML documents). We use it to see how LEILA performs on a corpus with a high structural heterogeneity. Since the querying was done automatically, the downloaded pages include spurious advertisements as well as pages with no proper sentences at all.

We tested LEILA on different target relations with increasing complexity:

- **birthdate:** This relation holds between a person and his birth date (e.g. "Chopin" / "1810"). It is easy to learn, because it is bound to strong surface clues (the first element is always a name, the second is always a date).
- **synonymy:** This relation holds between two names that refer to the same entity (e.g. "UN"/"United Nations"). The relation is more sophisticated, since there are no surface clues.
- **instanceOf:** This relation is even more sophisticated, because the sentences often express it only implicitly.

We compared LEILA to different **competitors**. We only considered competitors that, like LEILA, extract the information from a corpus without using other Internet sources. We wanted to avoid running the competitors on our own corpora or on our own target relations, because we could not be sure to achieve a fair tuning of the competitors. Hence we ran LEILA on the corpora and the target relations that our competitors have been tested on by their authors. We compare the results of LEILA with the results reported by the authors. Our competitors, together with their respective corpora and relations, are:

- **TextToOnto**<sup>3</sup>: A state-of-the-art representative for non-deep pattern matching. The system provides a component for the `instanceOf` relation and takes arbitrary HTML documents as input. For completeness, we also consider its successor Text2Onto (Cimiano and Völker, 2005a), although it contains only default methods in its current state of development.
- **Snowball (Agichtein and Gravano, 2000)**: A recent representative of the slot-extraction paradigm. In the original paper, Snowball has been tested on the `headquarters` relation. This relation holds between a company and the city of its headquarters. Snowball was trained on a collection of some thousand documents and then applied to a test collection. For copyright reasons, we only had access to the test collection (150 text documents).
- (Cimiano and Völker, 2005b) present a new system that uses context to assign a concept to an entity. We will refer to this system as the **CV-system**. The approach is restricted to the `instanceOf`-relation, but it can classify instances even if the corpus does not contain explicit definitions. In the original paper, the system was tested on a collection of 1880 files from the Lonely Planet Internet site<sup>4</sup>.

For the **evaluation**, the output pairs of the system have to be compared to a table of ideal pairs. One option would be to take the ideal pairs from a pre-compiled data base. The problem is that these ideal pairs may differ from the facts expressed in the documents. Furthermore, these ideal pairs do not allow to measure how much of the document content the system actually extracted. This is why we chose to extract the ideal pairs manually from the documents. In our methodology, the ideal pairs comprise all pairs that a human would understand to be elements of the target relation. This involves full anaphora resolution, the solving of reference ambiguities, and the choice of truly defining concepts. For example, we accept Chopin as instance of `composer` but not as instance of `member`, even if the text says that he was a member of some club. Of course, we expect neither the competitors nor LEILA to achieve the results in the ideal table. However, this methodology is the only fair way of manual extraction, as it is guaranteed to be system-independent. If  $O$  denotes the multi-set of the output pairs and  $I$  denotes the multi-set of the ideal pairs, then precision, recall, and their

harmonic mean  $F1$  can be computed as

$$recall = \frac{|O \cap I|}{|I|} \quad precision = \frac{|O \cap I|}{|O|}$$

$$F1 = \frac{2 \times recall \times precision}{recall + precision}$$

To ensure a fair comparison of LEILA to Snowball, we use the same evaluation as employed in the original Snowball paper (Agichtein and Gravano, 2000), the *Ideal Metric*. The Ideal Metric assumes the target relation to be right-unique (i.e. a many-to-one relation). Hence the set of ideal pairs is right-unique. The set of output pairs can be made right-unique by selecting the pair with the highest confidence for each first component. Duplicates are removed from the ideal pairs and also from the output pairs. All output pairs that have a first component that is not in the ideal set are removed.

There is one special case for the CV-system, which uses the Ideal Metric for the non-right-unique `instanceOf` relation. To allow for a fair comparison, we used the *Relaxed Ideal Metric*, which does not make the ideal pairs right-unique. The calculation of recall is relaxed as follows:

$$recall = \frac{|O \cap I|}{|\{x | \exists y : (x, y) \in I\}|}$$

Due to the effort, we could extract the ideal pairs only for a sub-corpus. To ensure significance in spite of this, we compute confidence intervals for our estimates: We interpret the sequence of output pairs as a repetition of a Bernoulli-experiment, where the output pair can be either correct (i.e. contained in the ideal pairs) or not. The parameter of this Bernoulli-distribution is the precision. We estimate the precision by drawing a sample (i.e. by extracting all ideal pairs in the sub-corpus). By assuming that the output pairs are identically independently distributed, we can calculate a confidence interval for our estimation. We report confidence intervals for precision and recall for a confidence level of  $\alpha = 95\%$ . We measure precision at different levels of recall and report the values for the best F1 value. We used approximate string matching techniques to account for different writings of the same entity. For example, we count the output pair "Chopin" / "composer" as correct, even if the ideal pairs contain "Frederic.Chopin" / "composer". To ensure that LEILA does not just reproduce the example pairs, we list the percentage of examples among the output pairs. During our evaluation, we found that the Link Grammar parser does not finish parsing on roughly 1% of the files for unknown reasons.

<sup>3</sup><http://www.sourceforge.net/projects/texttoonto>

<sup>4</sup><http://www.lonelyplanet.com/>

Table 1: Results with different relations

| Corpus          | Relation   | System     | #D  | #O  | #C  | #I   | Precision          | Recall             | F1     | %E    |
|-----------------|------------|------------|-----|-----|-----|------|--------------------|--------------------|--------|-------|
| Wikicomposers   | birthdate  | LEILA(SVM) | 87  | 95  | 70  | 101  | 73.68% $\pm$ 8.86% | 69.31% $\pm$ 9.00% | 71.43% | 4.29% |
| Wikicomposers   | birthdate  | LEILA(kNN) | 87  | 90  | 70  | 101  | 78.89% $\pm$ 8.43% | 70.30% $\pm$ 8.91% | 74.35% | 4.23% |
| Wikigeography   | synonymy   | LEILA(SVM) | 81  | 92  | 74  | 164  | 80.43% $\pm$ 8.11% | 45.12% $\pm$ 7.62% | 57.81% | 5.41% |
| Wikigeography   | synonymy   | LEILA(kNN) | 81  | 143 | 105 | 164  | 73.43% $\pm$ 7.24% | 64.02% $\pm$ 7.35% | 68.40% | 4.76% |
| Wikicomposers   | instanceOf | LEILA(SVM) | 87  | 685 | 408 | 1127 | 59.56% $\pm$ 3.68% | 36.20% $\pm$ 2.81% | 45.03% | 6.62% |
| Wikicomposers   | instanceOf | LEILA(kNN) | 87  | 790 | 463 | 1127 | 58.61% $\pm$ 3.43% | 41.08% $\pm$ 2.87% | 48.30% | 7.34% |
| Wikigeneral     | instanceOf | LEILA(SVM) | 287 | 921 | 304 | 912  | 33.01% $\pm$ 3.04% | 33.33% $\pm$ 3.06% | 33.17% | 3.62% |
| Googlecomposers | instanceOf | LEILA(SVM) | 100 | 787 | 210 | 1334 | 26.68% $\pm$ 3.09% | 15.74% $\pm$ 1.95% | 19.80% | 4.76% |
| Googlecomposers | instanceOf | LEILA(kNN) | 100 | 840 | 237 | 1334 | 28.21% $\pm$ 3.04% | 17.77% $\pm$ 2.05% | 21.80% | 8.44% |
| Googlec.+Wikic. | instanceOf | LEILA(SVM) | 100 | 563 | 203 | 1334 | 36.06% $\pm$ 3.97% | 15.22% $\pm$ 1.93% | 21.40% | 5.42% |
| Googlec.+Wikic. | instanceOf | LEILA(kNN) | 100 | 826 | 246 | 1334 | 29.78% $\pm$ 3.12% | 18.44% $\pm$ 2.08% | 22.78% | 7.72% |

#O – number of output pairs

#C – number of correct output pairs

#I – number of ideal pairs

#D – number of documents in the hand-processed sub-corpus

%E – proportion of example pairs among the correct output pairs

Recall and Precision with confidence interval at  $\alpha = 95\%$

## 3.2 Results

### 3.2.1 Results on different relations

Table 1 summarizes our experimental results with LEILA on different relations. For the **birthdate** relation, we used Edward Morykwas’ list of famous birthdays<sup>5</sup> as examples. As counterexamples, we chose all pairs of a person that was in the examples and an incorrect birthdate. All pairs of a proper name and a date are candidates. We ran LEILA on the Wikicomposer corpus. LEILA performed quite well on this task. The patterns found were of the form “X was born in Y” and “X (Y)”.

For the **synonymy** relation we used all pairs of proper names that share the same synset in WordNet as examples (e.g. “UN”/“United Nations”). As counterexamples, we chose all pairs of nouns that are not synonymous in WordNet (e.g. “rabbit”/“composer”). All pairs of proper names are candidates. We ran LEILA on the Wikigeography corpus, because this set is particularly rich in synonyms. LEILA performed reasonably well. The patterns found include “X was known as Y” as well as several non-grammatical constructions such as “X (formerly Y)”.

For the **instanceOf** relation, it is difficult to select example pairs, because if an entity belongs to a concept, it also belongs to all super-concepts. However, admitting each pair of an entity and one of its super-concepts as an example would result in far too many false positives. The problem is to determine for each entity the (super-)concept that is most likely to be used in a natural language definition of that entity. Psychological evidence (Rosch et al., 1976) suggests that humans prefer a certain layer of concepts in the taxonomy to classify entities. The set of these concepts is called the *Basic Level*. Heuristically, we found that the lowest super-concept in WordNet that is not a compound word is a good approximation of the ba-

sic level concept for a given entity. We used all pairs of a proper name and the corresponding basic level concept of WordNet as examples. We could not use pairs of proper names and incorrect super-concepts as counterexamples, because our corpus Wikipedia knows more meanings of proper names than WordNet. Therefore, we used all pairs of a common noun and an incorrect super-concept from WordNet as counterexamples. All pairs of a proper name and a WordNet concept are candidates.

We ran LEILA on the Wikicomposers corpus. The performance on this task was acceptable, but not impressive. However, the chances to obtain a high recall and a high precision were significantly decreased by our tough evaluation policy: The ideal pairs include tuples deduced by resolving syntactic and semantic ambiguities and anaphoras. Furthermore, our evaluation policy demands that non-defining concepts like *member* not be chosen as instance concepts. In fact, a high proportion of the incorrect assignments were *friend*, *member*, *successor* and *predecessor*, decreasing the precision of LEILA. Thus, compared to the gold standard of humans, the performance of LEILA can be considered reasonably good. The patterns found include the Hearst patterns (Hearst, 1992) “Y such as X”, but also more complex patterns like “X was known as a Y”, “X [...] as Y”, “X [...] can be regarded as Y” and “X is unusual among Y”. Some of these patterns could not have been found by primitive regular expression matching.

To test whether thematic heterogeneity influences LEILA, we ran it on the Wikigeneral corpus. Finally, to try the limits of our system, we ran it on the Googlecomposers corpus. As shown in Table 1, the performance of LEILA dropped in these increasingly challenging tasks, but LEILA could still produce useful results. We can improve the results on the Googlecomposers corpus by adding the Wikicomposers corpus for training.

<sup>5</sup><http://www.famousbirthdates.com>

The different learning methods (kNN and SVM) performed similarly for all relations. Of course, in each of the cases, it is possible to achieve a higher precision at the price of a lower recall. The runtime of the system splits into parsing ( $\approx 40s$  for each document, e.g. 3:45h for Wikigeography) and the core algorithm (2-15min for each corpus, 5h for the huge Wikigeneral).

### 3.2.2 Results with different competitors

Table 2 shows the results for comparing LEILA against various competitors (with LEILA in bold-face). We compared LEILA to **TextToOnto** and **Text2Onto** for the `instanceOf` relation on the Wikicomposers corpus. **TextToOnto** requires an ontology as source of possible concepts. We gave it the WordNet ontology, so that it had the same preconditions as LEILA. **Text2Onto** does not require any input. **Text2Onto** seems to have a precision comparable to ours, although the small number of found pairs does not allow a significant conclusion. Both systems have drastically lower recall than LEILA.

For **Snowball**, we only had access to the test corpus. Hence we trained LEILA on a small portion (3%) of the test documents and tested on the remaining ones. Since the original 5 seed pairs that Snowball used did not appear in the collection at our disposal, we chose 5 other pairs as examples. We used no counterexamples and hence omitted the Training Phase of our algorithm. LEILA quickly finds the pattern "Y-based X". This led to very high precision and good recall, compared to Snowball – even though Snowball was trained on a much larger training collection.

The **CV-system** differs from LEILA, because its ideal pairs are a table, in which each entity is assigned to its most likely concept according to a human understanding of the text – independently of whether there are explicit definitions for the entity in the text or not. We conducted two experiments: First, we used the document set used in Cimiano and Völker's original paper (Cimiano and Völker, 2005a), the Lonely Planet corpus. To ensure a fair comparison, we trained LEILA separately on the Wikicomposers corpus, so that LEILA cannot have example pairs in its output. For the evaluation, we calculated precision and recall with respect to an ideal table provided by the authors. Since the CV-system uses a different ontology, we allowed a distance of 4 edges in the WordNet hierarchy to count as a match (for both systems). Since the explicit definitions that our system relies on were sparse in the corpus, LEILA performed worse than the competitor. In a second experi-

ment, we had the CV-system run on the Wikicomposers corpus. As the CV-system requires a set of target concepts, we gave it the set of all concepts in our ideal pairs. Furthermore, the system requires an ontology on these concepts. We gave it the WordNet ontology, pruned to the target concepts with their super-concepts. We evaluated by the Relaxed Ideal Metric, again allowing a distance of 4 edges in the WordNet hierarchy to count as a match (for both systems). This time, our competitor performed worse. This is because our ideal table is constructed from the definitions in the text, which our competitor is not designed to follow. These experiments only serve to show the different philosophies in the definition of the ideal pairs for the CV-system and LEILA. The CV-system does not depend on explicit definitions, but it is restricted to the `instanceOf`-relation.

## 4 Conclusion and Outlook

We addressed the problem of automatically extracting instances of arbitrary binary relations from natural language text. The key novelty of our approach is to apply a deep syntactic analysis to this problem. We have implemented our approach and showed that our system LEILA outperforms existing competitors.

Our current implementation leaves room for future work. For example, the linkages allow for more sophisticated ways of resolving anaphoras or matching patterns. LEILA could learn numerous interesting relations (e.g. `country / president` or `isAuthorOf`) and build up an ontology from the results with high confidence. LEILA could acquire and exploit new corpora on its own (e.g., it could read newspapers) and it could use its knowledge to acquire and structure its new knowledge more efficiently. We plan to exploit these possibilities in our future work.

### 4.1 Acknowledgements

We would like to thank Eugene Agichtein for his caring support with Snowball. Furthermore, Johanna Völker and Philipp Cimiano deserve our sincere thanks for their unreserved assistance with their system.

### References

- [Agichtein and Gravano2000] E. Agichtein and L. Gravano. 2000. *Snowball*: extracting relations from large plain-text collections. In *ACM 2000*, pages 85–94, Texas, USA.
- [Brin1999] Sergey Brin. 1999. Extracting patterns and relations from the world wide web. In *Selected papers from the Int. Workshop on the WWW and Databases*, pages 172–183, London, UK. Springer-Verlag.
- [Buitelaar and Ramaka2005] P. Buitelaar and S. Ramaka. 2005. Unsupervised ontology-based semantic tagging

Table 2: Results with different competitors

| Corpus         | M | Relation     | System     | #D | #O  | #C  | #I   | Prec                  | Rec                   | F1            |
|----------------|---|--------------|------------|----|-----|-----|------|-----------------------|-----------------------|---------------|
| Snowball corp. | S | headquarters | LEILA(SVM) | 54 | 92  | 82  | 165  | <b>89.13%</b> ± 6.36% | <b>49.70%</b> ± 7.63% | <b>63.81%</b> |
| Snowball corp. | S | headquarters | LEILA(kNN) | 54 | 91  | 82  | 165  | <b>90.11%</b> ± 6.13% | <b>49.70%</b> ± 7.63% | <b>64.06%</b> |
| Snowball corp. | S | headquarters | Snowball   | 54 | 144 | 49  | 165  | 34.03% ± 7.74%        | 29.70% ± 6.97%        | 31.72%        |
| Snowball corp. | I | headquarters | LEILA(SVM) | 54 | 50  | 48  | 126  | <b>96.00%</b> ± 5.43% | <b>38.10%</b> ± 8.48% | <b>54.55%</b> |
| Snowball corp. | I | headquarters | LEILA(kNN) | 54 | 49  | 48  | 126  | <b>97.96%</b> ± 3.96% | <b>38.10%</b> ± 8.48% | <b>54.86%</b> |
| Snowball corp. | I | headquarters | Snowball   | 54 | 64  | 31  | 126  | 48.44% ± 12.24%       | 24.60% ± 7.52%        | 32.63%        |
| Wikicomposers  | S | instanceOf   | LEILA(SVM) | 87 | 685 | 408 | 1127 | <b>59.56%</b> ± 3.68% | <b>36.20%</b> ± 2.81% | <b>45.03%</b> |
| Wikicomposers  | S | instanceOf   | LEILA(kNN) | 87 | 790 | 463 | 1127 | <b>58.61%</b> ± 3.43% | <b>41.08%</b> ± 2.87% | <b>48.30%</b> |
| Wikicomposers  | S | instanceOf   | Text2Onto  | 87 | 36  | 18  | 1127 | 50.00%                | 1.60% ± 0.73%         | 3.10%         |
| Wikicomposers  | S | instanceOf   | TextToOnto | 87 | 121 | 47  | 1127 | 38.84% ± 8.68%        | 4.17% ± 1.17%         | 7.53%         |
| Wikicomposers  | R | instanceOf   | LEILA(SVM) | 87 | 336 | 257 | 744  | <b>76.49%</b> ± 4.53% | <b>34.54%</b> ± 3.42% | <b>47.59%</b> |
| Wikicomposers  | R | instanceOf   | LEILA(kNN) | 87 | 367 | 276 | 744  | <b>75.20%</b> ± 4.42% | <b>37.10%</b> ± 3.47% | <b>49.68%</b> |
| Wikicomposers  | R | instanceOf   | CV-system  | 87 | 134 | 30  | 744  | 22.39%                | 4.03% ± 1.41%         | 6.83%         |
| Lonely Planet  | R | instanceOf   | LEILA(SVM) | –  | 159 | 42  | 289  | <b>26.42%</b> ± 6.85% | <b>14.53%</b> ± 4.06% | <b>18.75%</b> |
| Lonely Planet  | R | instanceOf   | LEILA(kNN) | –  | 168 | 44  | 289  | <b>26.19%</b> ± 6.65% | <b>15.22%</b> ± 4.14% | <b>19.26%</b> |
| Lonely Planet  | R | instanceOf   | CV-system  | –  | 289 | 92  | 289  | 31.83% ± 5.37%        | 31.83% ± 5.37%        | 31.83%        |

M – Metric (S: Standard, I: Ideal Metric, R: Relaxed Ideal Metric). Other abbreviations as in Table 1

- for knowledge markup. In W. Buntine, A. Hotho, and Stephan Bloehdorn, editors, *Workshop on Learning in Web Search at the ICML 2005*.
- [Buitelaar et al.2004] P. Buitelaar, D. Olejnik, and M. Sintek. 2004. A protege plug-in for ontology extraction from text based on linguistic analysis. In *ESWS 2004*, Heraklion, Greece.
- [Califf and Mooney1997] M. Califf and R. Mooney. 1997. Relational learning of pattern-match rules for information extraction. *ACL-97 Workshop in Natural Language Learning*, pages 9–15.
- [Cimiano and Völker2005a] P. Cimiano and J. Völker. 2005a. Text2onto - a framework for ontology learning and data-driven change discovery. In A. Montoyo, R. Munozand, and E. Metais, editors, *Proc. of the 10th Int. Conf. on Applications of Natural Language to Information Systems*, pages 227–238, Alicante, Spain.
- [Cimiano and Völker2005b] P. Cimiano and J. Völker. 2005b. Towards large-scale, open-domain and ontology-based named entity classification. In *Int. Conf. on Recent Advances in NLP 2005*, pages 166–172.
- [Cimiano et al.2005] P. Cimiano, G. Ladwig, and S. Staab. 2005. Gimme the context: Contextdriven automatic semantic annotation with cpankow. In Allan Ellis and Tatsuya Hagino, editors, *WWW 2005*, Chiba, Japan.
- [Etzioni et al.2004] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2004. Web-scale information extraction in knowitall (preliminary results). In *WWW 2004*, pages 100–110.
- [Fellbaum1998] C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- [Finn and Kushmerick2004] A. Finn and N. Kushmerick. 2004. Multi-level boundary classification for information extraction. In *ECML 2004*, pages 111–122.
- [Freitag and Kushmerick2000] D. Freitag and N. Kushmerick. 2000. Boosted wrapper induction. In *American Nat. Conf. on AI 2000*.
- [Graupmann2004] Jens Graupmann. 2004. Concept-based search on semi-structured data exploiting mined semantic relations. In *EDBT Workshops*, pages 34–43.
- [Hearst1992] A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *ICCL 1992*, Nantes, France.
- [J. Iria2005] F. Ciravegna J. Iria. 2005. Relation extraction for mining the semantic web.
- [Maedche and Staab2000] A. Maedche and S. Staab. 2000. Discovering conceptual relations from text. In W. Horn, editor, *ECAI 2000*, pages 85–94, Berlin, Germany.
- [Mann and Yarowsky2005] Gideon Mann and David Yarowsky. 2005. Multi-field information extraction and cross-document fusion. In *ACL 2005*.
- [Montague1974] R. Montague. 1974. Universal grammar. In *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press.
- [Ravichandran and Hovy2002] D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL 2002*, Philadelphia, USA.
- [Riloff1996] E. Riloff. 1996. Automatically generating extraction patterns from untagged text. *Annual Conf. on AI 1996*, pages 1044–1049.
- [Rosch et al.1976] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Bream. 1976. Basic objects in natural categories. *Cognitive Psychology*, pages 382–439.
- [Ruiz-Casado et al.2005] Maria Ruiz-Casado, Enrique Alfonso, and Pablo Castells. 2005. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *NLDB 2006*, pages 67–79.
- [Sleator and Temperley1993] D. Sleator and D. Temperley. 1993. Parsing english with a link grammar. *3rd Int. Workshop on Parsing Technologies*.
- [Soderland et al.1995] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. Crystal: Inducing a conceptual dictionary. *IJCAI 1995*, pages 1314–1319.
- [Soderland1999] S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, pages 233–272.
- [Suchanek et al.2006] Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. In *SIGKDD 2006*.
- [Xu and Krieger2003] F. Xu and H. U. Krieger. 2003. Integrating shallow and deep nlp for information extraction. In *RANLP 2003*, Borovets, Bulgaria.
- [Xu et al.2002] F. Xu, D. Kurz, J. Piskorski, and S. Schmeier. 2002. Term extraction and mining term relations from free-text documents in the financial domain. In *Int. Conf. on Business Information Systems 2002*, Poznan, Poland.
- [Yangarber et al.2000] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *ICCL 2000*, pages 940–946, Morristown, NJ, USA. Association for Computational Linguistics.
- [Yangarber et al.2002] R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised learning of generalized names. In *ICCL 2002*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.