Mining the Web Through Verbs: A Case Study

Peyman Sazedj and H. Sofia Pinto

Inesc-ID
Rua Alves Redol 9, Apartado 13069 1000-029 Lisboa, Portugal
{psaz,sofia}@algos.inesc-id.pt

Abstract. Mining non-taxonomic relations is an important part of the Semantic Web puzzle. Building on the work of the semantic annotation community, we address the problem of extracting relation instances among annotated entities. In particular, we analyze the problem of verb-based relation instantiation in some detail and present a heuristic domain independent approach, based on verb chunking and entity clustering, which doesn't require parsing. We also address the problem of mapping linguistic tuples to relations from the ontology. A case study conducted within the biography domain demonstrates the validity of our results in contrast to related work, whilst examining the complexity of the extraction task and the feasibility of verb-based extraction in general.

1 Introduction

Unsupervised annotation tools have promoted a new perspective on harvesting data from the web, whilst presenting a convenient solution for populating knowledge bases with concept instances. Concept instances alone are not very expressive, unless accompanied by relations, therefore a lot of effort is being directed towards unsupervised extraction of non-taxonomic relations. There is an enormous amount of data on the web, requiring efficient extraction methods. Moreover the data is error-prone, requiring fault tolerant systems. Shallow extraction techniques are therefore clearly advantageous over more knowledge intensive methods, specially when they produce similar or even better results. Thus, the aim of this work is to present a shallow heuristic approach for mining relation instances through verbs, which achieves satisfactory precision and recall. The approach has been implemented within the FactBox framework [1], specifically designed for prototyping relation extraction algorithms and requires no parsing. Instead, it relies on verb-chunking and clustering to extract verb and entity pairs. A novel algorithm is presented, to match extracted relation candidates with relations from the ontology, whenever applicable. Being more efficient than other approaches based on dependency or full syntactic parsing, it is a more suitable candidate for mining large and error-prone data from the web. It also improves over the algorithms presented in [1] and compares to other state of the art algorithms which follow more knowledge intensive approaches.

Section 2 presents an overview of related work. Section 3 and 4 provide a generic problem description and a formal analysis of the task at hand. Section 5 discusses our algorithms. Section 6 presents the framework which served as a testbed for their evaluation, followed by a specific case study in section 7, the results of which are analyzed in section 8. Section 9 summarizes our main conclusions.

2 Related Work

Relation Instantiation falls within the category of Ontology Population, a newly emerging field within the Semantic Web community. Nevertheless, underlying techniques are often one and the same with those of the Information Extraction [2] and Question Answering [3,4] communities. A sister field, Ontology Learning [5,6], also has similar purpose and methods: to learn relations among concepts, instead of extracting relation instances among concept instances. One of the differences is that ontology learning puts a strong emphasis on statistical analysis, whereas in relation instantiation the balance is more inclined towards the linguistic component. In the following we discuss some relevant contributions for mining non-taxonomic relation instances from text.

One class of extraction systems are pattern based systems, with varying degrees of linguistic analysis, where patterns may be hand-crafted or automatically learned. A famous example of hand-crafted patterns are the Hearst patterns for extracting is-a relations [7]. Instead of using lexical patterns which hardly generalize over different domains, a popular alternative is to apply patterns to deeper syntactic or semantic structures. Some recent approaches use hand-crafted patterns to extract verb-based relations over dependency parse trees [8,9]. In [10] a generalization of the Hearst patterns is presented, based on an automatic pattern induction system. Similarly, and equally interesting, are pattern induction systems over dependency trees [3]. Even though the approach in [3] is targeted at question answering, it resembles relation instantiation, since the answer to a question often consists in querying relations among entities. DIPRE [11] and Snowball [12] are two relation extraction systems which learn patterns from seed tuples. Similarly, Armadillo [13] applies adaptive information extraction to learn extraction patterns.

Machine learning systems use a set of training examples, alike pattern induction systems, to train a classifier. A popular set of classifiers are based on kernel methods, and in particular, support vector machines. Kernel methods for relation extraction have been presented in [14]; [15] defines dependency tree kernels for learning relations from dependency parse trees. LEILA is a recent learning system which trains a classifier on link grammar structures [16]. A mixed approach is followed in [17], where probabilistic parse trees are augmented with entities and relations in order to train a classifier.

Yet another approach consists of transforming text into logical formula [18], where deduction can be used to extract more complex relations.

3 Generic Problem Description

Our goal is to develop a shallow method for extracting relation instances from web pages. Throughout the remainder of this paper, we assume that named entities of a text have been previously annotated with concepts from an ontology. In practical terms, annotations are produced by a third-party tool and are then read by the FactBox framework [1], whose main features will be reviewed in section 6. Consequently, the challenge lies in extracting those relations among entities, which have been modeled within the ontology. The problem is a very complex one, since different relations materialize through an unnumbered variety of linguistic expressions. Even a single relationship may often be expressed in so many ways, that we know of no single method for extracting all relations of a kind. Recent work [5,8] has shown that verbs are invaluable sources of evidence for retrieving relationships among entities, and methods such as dependency parsing have become popular for extracting verbs and their arguments. This work further explores the trend of verb-based extraction methods and aims to address the problem of relating verb arguments, without parsing, and selecting those verb-entity pairs which match the semantics of the ontology.

3.1 Divide and Conquer

Ontology-based extraction of relation instances from verbs can be divided into two sub-problems: (1) the selection of verb arguments (e.g. entity pairs in case of binary relations) and (2) the mapping of verbs with appropriate relations from the ontology. Both sub-problems are dependent parts of one whole, resembling a constraint satisfaction problem, where the solution of one restricts the solution space of the other. Nevertheless, they can be solved in any order, and the choice is relevant as we will show. Before we analyze each of the problems in greater detail, consider the example sentence "John works at IBM and likes Jill", where John and Jill are instances of `person` and IBM of `company`. Assume that the ontology connects the classes `person` and `company` with two relations, namely `employee` and `investor`. Problem (1) consists of selecting the correct arguments of each verb and would result in the two potential relation candidates `works(John, IBM)` and `likes(John, Jill)`. Problem (2), on the other hand, consists of mapping verbs to relations from the ontology, in this case, to map the verb `works` to the relation `employee`. This two-step process would result in the extraction of `employee(John, IBM)`. If we start out by determining that John and IBM are related, the problem is simplified to that of discovering whether any of `employee` or `investor` relations hold. On the other hand, if we had started by discovering that the verb `works` matches the `employee` relation, the argument selection process would have become restricted to choosing whether John or Jill work at IBM. In the following we analyze each of the problems in greater detail.

3.2 Selecting Verb Arguments

The most naive solution for this problem is to randomly map each verb with all entities in some vicinity. If the vicinity is sufficiently large, at maximum having the size of the whole text, it is guaranteed that all potential matches will be found. This approach makes little sense of course, and yields factorial complexity $O(n!)$[1]. To simplify argument selection, two restrictions are often

[1] To simplify we assume there are no reflexive relations.

implicitly adopted: **a locality restriction**, only considering arguments in a local vicinity of the verb (e.g. a part of the sentence); **an arity restriction**, focusing on binary relations (since n-ary relations can be decomposed into n binary relations). The first restriction substantially reduces the search space of potential entities, whereas the second restriction reduces the complexity of the task to $O(n^2)$. In an ontology-based setup, as in many schema-based setups, a third restriction applies: **a type restriction** where the arguments of a relation must be of a specified type, in other words, only some entity types potentially match to form relation candidates, filtering out incompatible entity pairs (those which are not related within the ontology). Complexity is reduced to $O(|i|\,|j|)$, where the relation holds among instances of classes i and j. Nevertheless, the worst case complexity is still of quadratic order.

3.3 Mapping Verbs to Relations

This problem is sometimes solved by using a lexicon which directly maps lexical elements (such as verbs) to relations from the ontology. Although convenient in situations where such a lexicon is available, it requires a large amount of work to build one, specially when mining heterogeneous data from different domains. We aim to develop a domain-independent strategy which tries to map verbs to relations based on linguistic evidence. For that purpose, we may roughly distinguish three degrees of linguistic analyses: (1) a lexical analysis which tries to establish a lexical equivalence among the verb and the relation name; (2) a shallow semantic analysis which aims to establish an intentional equivalence among the verb and the relation name (e.g. by considering synonyms); and (3) a deep semantic analysis, which tries to match the selectional restrictions of the verb with the constraints of a relation. In this work we focus on the first two degrees - a lexical and a shallow semantic approach implemented within the FactBox framework. The approach consists of locating a verb and a relation name within the WordNet hierarchy, and deciding whether they are sufficiently related or not.

4 Formal Analysis

We define an ontology as a set of concepts C ordered by a subsumption relation within a taxonomy.[2] We consider a set of labeled non-taxonomic binary relations R among concepts of C, denoted as $r(c,d)$, $r \in R \wedge c,d \in C$. Entities in the text are considered instances of concepts and denoted by a set I. For an entity pair $i,j \in I$ and two classes $c,d \in C$ of which i and j are instances, let us consider a relation selector $S_{rel} : C \times C \rightarrow R \cup \{\perp\}$, which, given two concepts c and d, returns a subset of R, augmented with the empty relation $\perp$, which can be interpreted as the set of potential relations among c and d which are not modeled within the ontology.

[2] We have omitted the formal definition of the subsumption relation for the sake of simplicity.

As explained in section 3.2, the argument selection task, S, has a worst case time complexity of $O_S(n^2)$ for each verb. This translates into a total complexity of $O_S(n^2\,|V|)$, considering V the set of all verb occurrences of a corpus. The mapping task, M, has a worst case complexity of $O_M(|V|\,|R|)$. In order to understand how the two sub-problems affect each other and the total problem complexity, let us consider both solutions SM and MS, depending on the order in which the problems are solved. In SM (M is solved after S), only a limited subset of relations will have to be considered and final complexity is $O_{SM}(n^2\,|V|\,|S_{rel}|)$. If the tasks are solved in inverted order, we obtain a complexity of $O_{MS}(|V|\,|R|\,|i|\,|j|)$, where $|i|$ and $|j|$ are the number of entities of type i and j, applicable to the relations picked by M.

A few observations follow. The locality restriction of verbs (section 3.2) says that only arguments within some local vicinity of a verb are plausible candidates. Let E be the set of entities within the vicinity of a verb, then $|E| < \epsilon_1$, for some constant ϵ_1. Likewise, $|S_{rel}| < \epsilon_2 < |R| + 1$. A particularity is that both ϵ_1 and ϵ_2 are constants, whereas $|V|$ grows with the size of the corpus. It follows that $|V| >> \epsilon_1, \epsilon_2$, for a large enough corpus, therefore the time complexity can be rewritten as $O(m\epsilon_1^2\epsilon_2)$, where $m = |V|$. Thus, in reality the problem is of linear complexity. In a similar way it can be shown that the same holds for O_{MS}.

The question remains which one of the two is more efficient. Since the number of entities within the vicinity of a verb is always less than a threshold ϵ_1, the answer lies within the complexity of the ontology. For a large ontology, $|R|\,/\,|S_{rel}| > \epsilon_1^2$, and SM is the preferred method. The more specific the relations of the ontology are, the smaller is $|S_{rel}|$ in contrast to $|R|$, and the better the performance of SM.

5 Algorithms

Based on the division of the general problem into two sub-problems (section 3.1), we present distinct algorithms, for the selection of verb arguments and for mapping verbs to relations.

5.1 Selecting Verb Arguments

Regarding the selection of verb arguments, we used a simple verb chunker to identify verb groups. For each verb group, the challenge consists of discovering the syntactic arguments of the verb, without parsing. As a first approach (S_1) we consider all entities of a sentence as potential arguments of a verb, thus every possible entity pair of a sentence is matched with each verb of the sentence. We are implicitly restricting the vicinity of a verb to the lexical elements of a sentence, while ignoring verb anaphora, where the arguments of the same verb instance can span several sentences. Since S_1 yields a 100% recall in finding verb arguments, it is a good baseline to compare with more complex approaches.

$$S_1 : E \rightarrow \mathcal{P}(V) = V_s, V_s \subseteq V$$

V_s denotes the set of verbs of a sentence and is an element of the powerset of V. Since a sentence often contains several verbs, we developed a second approach (S_2) which further restricts the vicinity of verbs by dividing the sentence into segments and centering each segment around a verb. In other words, we employ clustering, where the number of clusters is equivalent to the number of verbs within the sentence and each verb is the center of a cluster. For each entity $e \in E$ of the sentence, the challenge consists of assigning it to the correct verb cluster $v \in V$. [3]

$$S_2 : E \rightarrow V = \arg\min_{v \in V} distance(e, v)$$

We define a distance function, $distance : E \times V \rightarrow \mathbb{N}$, which represents the numerical distance between an entity and the center of a verb cluster. Several metrics are possible for such a distance function, based on the linguistic evidence that is considered. We experiment with two metrics (short d_1 and d_2): (1) the distance between two words is given by the number of characters in between them, and (2) the distance is given by the number of words in between them. For the sentence "John works for IBM but loves Sisco.", $d_1(IBM, works) = 5$ and $d_1(IBM, loves) = 11$, whereas $d_2(IBM, works) = 1$ and $d_2(IBM, loves) = 2$. Moreover $d_2 \leq d_1$, because the number of words is always less than the number of characters. The main characteristic of d_2 over d_1 is that it disregards the length of words and is expected to have a more constant behavior in presence of both short and lengthy words.

5.2 Mapping Verbs to Relations

Let us recall that the problem consists of mapping verbs to relations from an ontology (for a more detailed description see section 3.3). We define a mapping function as a function M which maps a verb $v \in V$ to a relation $r \in R$ from the ontology.

$$M : V \rightarrow R \cup \{\perp\} = \begin{cases} \arg\max_{r \in R} similarity(v, r) \geq \theta, \theta \in [0, 1] \\ \perp, \text{otherwise.} \end{cases}$$

M returns the relation that maximizes a normalized similarity function defined as $similarity : V \times R \rightarrow [0, 1]$ and returns no mapping ($\perp$), if the maximum similarity between a verb and a relation is below a threshold θ.

As a first approach (M_1), we try to establish a mapping by means of string matching, based on a normalized Smith-Waterman distance [19].

We also define a shallow semantic approach (M_2), which queries WordNet in order to detect whether lexically dissimilar verbs and relation names are similar or equivalent in meaning, such as synonyms. For example, we may need to match the compound verb "was born" with a relation from the ontology which may

[3] Within a verb cluster, the semantic restrictions of the ontology allowed us to select entities in the correct order, without the need for additional criteria.

have been named "born", "birthday" or "date of birth". Likewise, suppose we are mining death dates. It is desirable to match the verbs "deceased" or "passed away" with a relation such as "date of death". More concretely, we are looking for a strategy to assess the similarity of verbs with verbs and verbs with nouns, whether simple or compound. We start with simple verbs and nouns, then we generalize for the compounds.

For matching two simple verbs, we simply check whether they belong to the same synset. We don't use a criterion based on the subsumption hierarchy of verbs as commonly done with nouns, because the verb hierarchy is very shallow and could potentially cause many false positives. For matching a simple verb with a simple noun, we measure the similarity among the derivationally related nouns of the verb with the noun. Thus, the task remains of measuring the relatedness of simple nouns, a well studied problem which may be solved by calculating the path from the root to the lowest common node that subsumes both nouns. Finally, the compounds, namely compound verbs and nouns remain. The main verb of a compound stands always in last position, therefore it is straightforward to filter out the main verb.[4] A compound noun, on the other hand, can be split into a set of simple nouns. Then, we filter nouns which do not exist in the WordNet hierarchy and those which characterize the domain of the relation and not the relation itself. For example a relation named "death date" has a date as its domain, therefore the noun "date" can be filtered out. We obtain a final set of nouns, each of which is compared individually with its counterpart, whether verb or noun. The efficiency of the previous method is not of great concern, because relation names within the ontology only need to be processed once. This can be done conveniently as a pre-processing step, before the extraction commences.

To summarize, consider V the set of simple verbs and N the set of simple nouns within WordNet. Consider V' and N' the sets of all possible compound verbs and nouns, respectively. A compound verb has a main verb $v \in V$, so that $\exists v_{\in V}, last(v') = v, v' \in V'$ returns the main verb. Further, consider the functions $synset(v), v \in V$, which returns the synset of a verb and $deriv(v), v \in V$, which returns the derivationally related nouns of a verb. Finally, consider the following functions: $sim(n_i, n_j) \in [0,1]$, calculates the normalized similarity among two nouns, $split(n) = X, X \cap N \neq \{\}$, splits a compound noun into its simple parts and $filter(X) = X', X' \subseteq X \cap N$, filters irrelevant nouns.

Our WordNet based similarity algorithm (Sim_{Wn}) is hence defined as follows:

$$Sim_{Wn} : x \times y \rightarrow [0,1] = \begin{cases} 1 & x \in synset(y) \wedge x, y \in V \\ sim(x,y) & x, y \in N \\ sim(last(x), y) & x \in V' \\ \max_{\forall n, n \in deriv(x)} sim(n,y) & x \in V \wedge y \in N \\ \max_{\forall n, n \in filter(split(y))} sim(x,n) & y \in N' \\ 0 & x, y \notin V \cup V' \cup N \cup N' \end{cases}$$

[4] This observation applies, at least, to the verb chunker we used.

6 FactBox Framework

FactBox is a testbed for relation instantiation algorithms and supports ontology population. It is composed of three plugin-based components: (1) an Extraction Component, (2) a Relation Base and (3) a Similarity Matcher. The Extraction Component is divided into an entity and a relation extraction component. Annotations of named entities can be produced by a third-party tool, given that a plugin is provided which reads the annotations and populates the Entity Base of the system. Entities are then read by relation extraction algorithms which populate the Relation Base with relation candidates. The Relation Base contains a rule chain which is capable of adding, retracting and transforming relation candidates by applying deductive rules (similar to implications of FOL). Finally, the Similarity Matcher tries to match instances of the Relation Base with relations from an ontology and populates a knowledge base with relation instances. Given a gold standard, the system performs automatic evaluation of the relation extraction algorithms over a corpus.

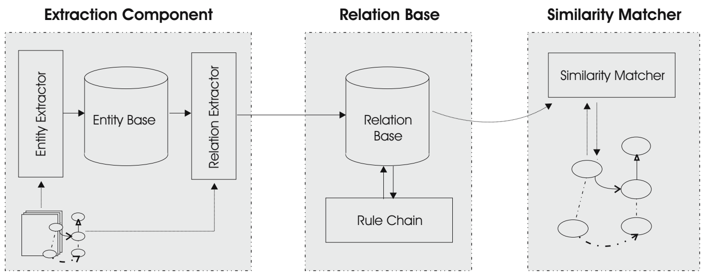

Fig. 1. The conceptual architecture of FactBox and its three main components

Figure 1 illustrates the architecture of the framework. The algorithms for selection of verb arguments (section 5.1) have been implemented as relation extraction plugins and the mapping algorithms (section 5.2) as plugins of the Similarity Matcher. For additional information on the framework, see [1].

7 Case Study: Biographies from IMDb

One of the challenges of ontology based relation extraction is to find appropriate corpora and ontologies for testing systems. This limitation has led some researchers to evaluate their systems on different domains, making comparison of results difficult; it has led others to work on a very small set of relations of a particular domain. In choosing a domain of interest, we attempted to pick a domain which is generally interesting for the public and can be reused in other experiments. We also tried to pick a domain which had already been tackled by other researchers. As another requirement, it should present the researcher with a wealth of possible relations. The last requirement was that the corpus should be as large as possible, preferably having been written by different people, in order to cover as many forms of expressing a relation as possible. The aforementioned conditions led us to create a corpus of biographies.[5] In this case study, we aim to explore the practical complexity of the extraction task, the performance of our approach, and the coverage of verbs with regard to other part of speech that may indicate the same relationship. In the following, we present the corpus, explain how it was acquired, describe the relations to be extracted and present some useful statistics on the corpus.

7.1 The IMDb Biography Corpus

The Internet Movie Database (IMDb) is a popular site for looking up information on movie reviews.[6] The site also offers a wealth of information about people involved with movies, including biographies of actors, directors and producers. Information on IMDb is a result of social collective effort. Having been written by thousands of people, it is reasonable to expect that biographies reflect different writing styles, rendering this corpus particularly interesting for the proposed analysis. Additionally, the IMDb site also offers some information in semi-structured form, such as date and place of birth or death. This information is very useful for automatically mining a golden standard for those relations. Crawling the IMDb site for biographies, we obtained a corpus of 2695 documents.[7]

7.2 Relations of Interest

From among the many possible relations of interest within the biography domain, we had to pick a few. Although our corpus is domain-specific, covering biographies of the movie domain, we aim to extract relations which are general enough to be applied to any biography. This choice is based on two reasons: (1) to ease comparison with other works, (2) because the annotation tool we used for marking-up named entities, by default, only recognizes `people`, `organizations`, `dates` and `locations`, making it difficult to work with domain-specific relations.

Thus, we came up with the following relations of interest: `birthday`, `birthplace`, `death date`, `death place`, `spouse` and `study`. Our ontology containing the aforementioned concepts and relations is illustrated in figure 2. The ontology was formalized in OWL. The `birthday` and `death date` relations were designed as functional relations, to ensure that each person had unique birth and death dates.

7.3 Statistics

Table 1 summarizes useful statistical estimates for the corpus, which were extracted from a random sample of 50 biographies. The first row summarizes the

[5] http://www.inesc-id.pt/˜psaz/

[6] http://www.imdb.com

[7] At some point we ended the crawling process; 2695 is *not* the total number of biographies available at IMDb.

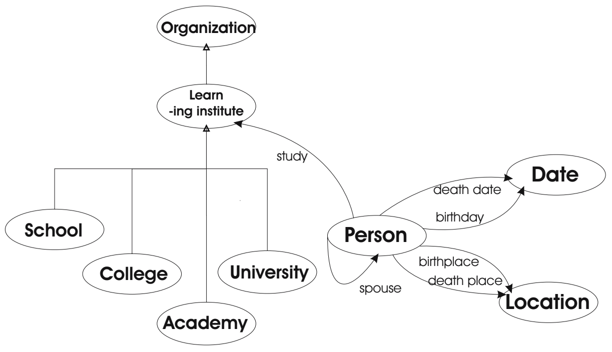

Fig. 2. The conceptual model of the biography ontology

number of instances of each relation without anaphora resolution, that is, disregarding pronouns as possible verb arguments. The second row considers only those relations where pronoun anaphora occurred, and row three sums up the total number of verb instances for each kind of relation. The next row is an estimate of the number of relation instances that occur per document, given in percentages. Finally, we show how often a relation occurs in other part of speech, apart from verbs, and calculate the coverage of verbs within the last row.

Table 1. Sample statistics and rough estimates for the corpus. VI stands for Verb Instances.

	birthday	birthplace	study	spouse	death date	death place
VI	26	53	7	3	4	4
VI with anaphora	1	5	21	14	10	4
Total VI	27	58	28	17	14	8
VI p/ document	0.52	1.1	0.14	0.06	0.08	0.08
Non VI	0	0	0	16	1	0
VI coverage (%)	100	100	100	52	93	100

A number of conclusions follows from the data in table 1. Only the `birthday` and `birthplace` relations exist abundantly, without considering anaphora. This is justified by the fact that a person's name is usually referred to in the beginning of a biography, jointly with date and place of birth, whereas future references to the person are made with pronouns. With anaphora resolution, the `study`, `spouse` and `death date` relations also occur sufficiently often. Regarding dates and places of death, they appear with low frequency, since most of the covered actors are still alive.

A major point under study is the coverage of verbs with regard to other part of speech. The data clearly shows the selected relations are mostly covered by verbs, except for the `spouse` relationship. The estimates hint that verb-based extraction methods promise to perform well on many domains such as the one under evaluation.

Table 2. Different modes of expressing the same relation

birthday	**study**
<person> was born on <date>	<person> was educated at <school>
Born on <date>, <person>	<person> majored at <school>
<person> was born ... on <date>	<person> attended <school>
birthplace	<person> was transferred to <school>
<person> was born in <location>	during ... at <school>, <person>
Born in <location>, <person>	After graduation at <school>, <person>
<person> was born ... in <location>	<person> graduated from <school>
<person> was born ... in <location>, <location>	**death date**
spouse	<person> died on <date>
After <person>'s divorce from <person>	On <date>, <person> was killed by
<person> married <person>	<person> was found dead ... <date>
<person> remarried <person>	**death place**
his wife <person>	<person> died in <location>
her husband <person>	<person> passed away ... <location>

From among the different verbal expressions we encountered in the sample, we selected a few to illustrate different ways of expressing each relation. The patterns are summarized in table 2.

8 Evaluation

It is extremely difficult to evaluate extraction systems on large corpora. Either a gold standard is somehow available for the entire corpus, or a sample has to be selected for which a reference is manually created. The work in [12] follows an interesting approach, automatically mining a reference standard from structured data, whilst manually creating a reference standard for a smaller subset of their corpus. They argue that both solutions are interesting. The former allows to gain valuable insight into large-scale extraction, which is often the actual aim of the work, while the latter allows to fine-tune results by detecting named-entity tagging errors. For obvious reasons, the two approaches may differ considerably in their results.

An overall important detail is to distinguish among *a priori* and *a posteriori* evaluations [5,20]. In an *apriori* evaluation, the gold standard is created before hand and the results of the system are measured against the reference. In an *a posteriori* evaluation, results are presented to an evaluator, who then decides which of them are correct and which are not. Comparing the strict a priori method with the more relaxed a posteriori, the work in [5] reports a degradation of about 10% in the precision of the former over the latter.

In our case, the FactBox framework is provided with a gold standard in digital format and the system automatically produces an *a priori* evaluation without human intervention. Following the approach of [12], we attempted to create two gold standards, one manually over a small subset of documents (section 8.1), and the other automatically over a larger subset (section 8.2).[8] Both experiments are described in the following. It is important to note that we used an unsupervised

[8] Structured information for automatically creating a gold standard was only available for a subset of the corpus.

named entity tagger and verb chunker, plugins of the Gate architecture [21], therefore a degradation of 10-20% is expected in all experiments due to tagging errors.

In both experiments, evaluation scores are presented separately for each relation, in order to convey deeper insight on the performance of each. Standard Precision, Recall and F1 metrics are defined as

$$P = \frac{|R_{es} \cap G|}{|R_{es}|}, R = \frac{|R_{es} \cap G|}{|G|}, F1 = \frac{2PR}{P+R}$$

where R_{es} is the result set (the set of instances that were mined) and G is the gold standard set.

8.1 Experiment 1

We randomly selected a sample of 50 documents from the corpus and manually created a gold standard for the sample. Recall that in section 5.1 we defined three algorithms for selecting verb arguments, a baseline S_1 and an improvement over the baseline which was defined based on two different distance metrics, $S_{2.d1}$ and $S_{2.d2}$. We compare the performance of the three algorithms with the `birthday` and `birthplace` relations. Since these two relations are those which occur more frequently, they are appropriate candidates. Additionally, both relations always materialize through the verb "born", meaning that the performance of the verb mapping algorithm can be discarded and the results do accurately reflect the performance of the verb argument selection algorithms. Table 3 summarizes our results.

Table 3. Comparison of different verb selection algorithms defined in section 5.1

	S_1	$S_{2.d1}$	$S_{2.d2}$
Precision (%)	58.5	75.4	75.8
Recall (%)	77.5	69.0	70.4
F1 (%)	66.7	72.1	73.0

The results are quite interesting and a few conclusions follow. First, we remark that the baseline which was supposed to obtain 100% recall, only obtains 77.5% recall. It is highly likely that the 22.5% decline in recall is due to name entity and verb chunking tagging errors. Thus, we should bear in mind that all results presented in the remainder of this paper may actually be improved up to approximately 20% if better taggers are used. Moreover, both distance metrics improve over the baseline and algorithm $S_{2.d2}$, based on a word distance metric, works best. In the remainder of our evaluations we use this algorithm.

In the following we inspect the performance of our approach for each kind of relation. Evaluation scores are presented in table 4.

The `birthday` and `birthplace` relations obtain very high precision and recall, given that the values include up to 20% of tagging errors. Our results are similar to those obtained by the machine learning approach in [16], who trained

Table 4. Evaluation scores for each kind of relation

	birthday	birthplace	study	spouse	death date	death place	study(2)	marry
Precision (%)	77.3	75.0	0	0	50.0	21.4	66.7	25.0
Recall (%)	73.9	68.8	0	0	50.0	75.0	57.1	33.3
F1 (%)	75.6	71.7	0	0	50.0	33.3	61.5	28.6

a classifier on the `birthday` relation and obtained precision and recall in the range of 70-80%. It is unclear whether their approach included tagging errors or not.

Regarding the `study` and `spouse` relations, we found that both of them had unhappy designations, since our mapping algorithm failed to match them with corresponding verbs. In table 2 one can see that the `study` relation is not expressed in a consistent way; very loose semantic associations between people and schools are sufficient to express this relation. Therefore, we implemented a transformation rule within FactBox, which transformed relations between people and schools into `study` relations. Regarding the `spouse` relation, we changed its name to `marry`. The last two columns of table 4 reflect the improvements that followed from these simple adaptations.

8.2 Experiment 2

In a second experiment, we evaluated the scalability of our approach by using larger samples. For that purpose, we automatically created a gold standard by mining structured data from the IMDb web site. This data is available for the `birthday`, `birthplace`, `death date`, `death place` and `spouse` relations. The mined data for each relation was only considered suitable, to be included in the gold standard of a biography, if it actually occured within the biography. This is so, because data available in structured form was often not mentioned within the biographical text. Unfortunately, this approach has a pitfall, since it does not exclude relations whose entities are referred to by pronouns. Since we do not have an automatic anaphora resolution module yet, only the `birthday`

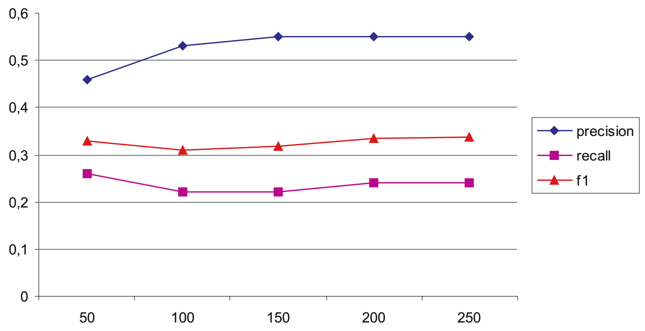

Fig. 3. Precision, Recall and F1 curves for different sample sizes

and `birthplace` relations are considered in this experiment, since they do not frequently occur with pronouns. Results are shown in figure 3.

We started with a sample of 50 documents and gradually increased the sample size. The recall values are very low in this experiment due to a faulty construction of the gold standard. In particular, having automatically obtained the place of birth of an actor, we check whether it is mentioned in the biography in order to decide whether to include it in the gold standard or not. This seems to be an unreliable approach, since the place of birth is often mentioned with some other intention, not expressing the actual `birthplace` relationship. Nevertheless, the experiment shows that the output of our approach remains stable over large samples, indeed results improved with larger samples until they converged.

9 Concluding Remarks

This work is a contribution towards large scale relation extraction from the web. We analyzed the problem of verb-based relation extraction and divided it into two sub-problems, leading to two novel algorithms: an unsupervised heuristic approach which performs verb-entity clustering, and an algorithm for mapping verbs to relations from an ontology. In a first experiment we showed that verb-based relation extraction is a feasible solution for mining relations from the web, and that, our approach in particular, compares to other state of the art algorithms that require more complex techniques. In a second experiment we tested the scalability of our approach, showing that it converges over larger samples. We also described the problems we encountered, namely, that verb arguments are often referred to by pronouns, requiring anaphora resolution to increase recall up to 5 times for of some relations. Future work includes the development of an efficient anaphora resolution module to be integrated within the FactBox framework and an improvement of current algorithms.

References

1. P. Sazedj and H. S. Pinto, FactBox - a Framework for Instantiating Ontological Relations from Text, in *Workshop on Web Content Mining with Human Language Technologies at ISWC*, 2006.
2. E. Marsh and D. Perzanowski, MUC-7 Evaluation of IE Technology: Overview of Results, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html.
3. D. Lin and P. Pantel, DIRT - Discovery of Inference Rules from Text, in *Proceedings of KDD*, pp. 323–328, 2001.
4. D. Ravichandran and E. H. Hovy, Learning surface text patterns for a Question Answering System, in *ACL*, pp. 41–47, 2002.
5. A. Schutz and P. Buitelaar, RelExt: A Tool for Relation Extraction from Text in Ontology Extension, in *Proceedings of ISWC*, pp. 593–606, 2005.
6. A. Maedche and S. Staab, Discovering Conceptual Relations from Text, in *Proceedings of ECAI*, pp. 321–325, 2000.
7. M. A. Hearst, Automatic acquisition of hyponyms from large text corpora., in *COLING*, pp. 539–545, 1992.

8. M. Ciaramita, et al, Unsupervised Learning of Semantic Relations between Concepts of a Molecular Biology Ontology, in *Proceedings of IJCAI*, pp. 659–664, 2005.
9. L. Specia and E. Motta, A Hybrid Approach for Relation Extraction Aimed at the Semantic Web, in *Proceedings of FQAS*, pp. 564–576, 2006.
10. R. Snow, D. Jurafsky, and A. Y. Ng, Learning Syntactic Patterns for Automatic Hypernym Discovery, in *NIPS*, 2004.
11. S. Brin, Extracting Patterns and Relations from the World Wide Web, in *WebDB*, pp. 172–183, 1998.
12. E. Agichtein, *Extracting Relations From Large Text Collections*, Ph.D. thesis, Columbia University, 2005.
13. F. Ciravegna, et al, Learning to Harvest Information for the Semantic Web, in *Proceedings of ESWS*, pp. 312–326, 2004.
14. D. Zelenko, C. Aone, and A. Richardella, Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, volume 3, pp. 1083–1106, 2003.
15. A. Culotta and J. S. Sorensen, Dependency tree kernels for relation extraction., in *ACL*, pp. 423–429, 2004.
16. F. M. Suchanek, G. Ifrim, and G. Weikum, Combining linguistic and statistical analysis to extract relations from web documents, in *KDD*, pp. 712–717, 2006.
17. S. Miller, et al, A Novel Use of Statistical Parsing to Extract Information from Text, in *ANLP*, pp. 226–233, 2000.
18. D. I. Moldovan and V. Rus, Logic Form Transformation of WordNet and its Applicability to Question Answering, in *ACL*, pp. 394–401, 2001.
19. T. F. Smith and M. S. Waterman, Identification of common molecular subsequences. *Journal of Molecular Biology*, volume 147, pp. 195–197, 1981.
20. P. Cimiano, M. Hartung, and E. Ratsch, Finding the Appropriate Generalization Level for Binary Relations Extracted from the Genia Corpus, in *LREC*, pp. pp. 161–169, 2006.
21. H. Cunningham, R. J. Gaizauskas, and Y. Wilks, A General Architecture for Language Engineering (GATE) - a new approach to Language Engineering R&D, Technical Report, Dept. of Computer Science, University of Sheffield, 1996.