Acta Cybernetica **19** (2010) 677–694.

Information Extraction from Wikipedia Using Pattern Learning

Márton Miháltz*

Abstract

In this paper we present solutions for the crucial task of extracting structured information from massive free-text resources, such as Wikipedia, for the sake of semantic databases serving upcoming Semantic Web technologies. We demonstrate both a verb frame-based approach using deep natural language processing techniques with extraction patterns developed by human knowledge experts and machine learning methods using shallow linguistic processing. We also propose a method for learning verb frame-based extraction patterns automatically from labeled data. We show that labeled training data can be produced with only minimal human effort by utilizing existing semantic resources and the special characteristics of Wikipedia. Custom solutions for named entity recognition are also possible in this scenario. We present evaluation and comparison of the different approaches for several different relations.

Keywords: natural language processing, information extraction, machine learning

1 Introduction

Today, in the world of the knowledge-hungry applications, there is an increased need for mass quantities of structured information that can enable searching technologies that go beyond simple character-based solutions. The construction of such Semantic Web technologies requires efforts to be made in the direction of automatically extracting semantic relations and properties of entities from available online textual resources. The work described here was carried out within the framework of the iGlue project[1], which aims to create a uniformly treated, semantically interlinked database of named entities such as persons, geographical names, institutions etc.

Recently, much attention was given to exploiting available large-scale online resources for information extraction, in particular, to using Wikipedia, the free-

*Péter Pázmány Catholic University, E-mail: mihaltz@digitus.itk.ppke.hu

[1] http://iglue.com

content online encyclopedia[2] ([3], [8], [14], [23], [26], [25]). The reasons we adopted Wikipedia for information extraction are that it has considerable coverage, the articles have good text quality making it possible to use state-of-the-art natural language processing algorithms, and finally because the redundancy between the structured and unstructured (free text) parts provides possibilities for generating annotated training data. Other special properties of Wikipedia pages (uniform encyclopedic structure, internal links, redirection pages etc.) provide further possibilities for enhancing information extraction.

Our research focuses on the development of a large-scale, reliable information extraction system, which mines for structured textual information, such as properties and relationships of entities, from the free text sections of the English pages of Wikipedia. Our system is therefore able to obtain information that is inaccessible from the structured sections (infoboxes, tables, category labels etc.) of the article pages. To leverage the task of processing free text, our system relies on natural language processing tools such as syntactic parsing, named entity recognition and coreference resolution.

Our system uses templates, or frames, consisting of slots that correspond to the entities that are in the given relation. The frame slots can be filled using extraction patterns, which are described using verb frame structures that have elements corresponding to the argument and modifier noun phrases of the main verbs in the input sentences.

In order to assess the performance of the linguistic analysis and to explore the potentials of the verb frame-based information extraction approach, we first created a system that uses hand-crafted extraction patterns. This system also served as a baseline for comparison to further research, in which we investigated a method to automatically learn extraction patterns. We also experimented with approaches that use less sophisticated linguistic analysis and machine learning.

The rest of this paper is organized as follows. In the next Section, we give an overview of related work. In Section 3, we describe the baseline system that uses manually crafted extraction patterns, the details of the linguistic analysis and the problems we encountered and solved. In section 4, we describe an alternative approach, using supervised machine learning. In section 5, we describe our pattern learning algorithm and compare its performance against both the baseline system and the machine learning approaches. Section 6 summarizes and discusses our results.

2 Related Work

Learning extraction patterns. Several authors have explored the possibilities of reducing the burden of manually authoring extraction patterns. Riloff [16] uses manual annotation only to categorize documents as relevant and irrelevant to generate extraction patterns. Another line of research was initiated by DIPRE [6], which relies on the redundancy of information on the web. Facts being expressed

[2] `http://en.wikipedia.org/wiki/`

in different forms enable bootstrapping from a small number of manually supplied seeds. The Snowball system [2] uses term vectors to represent the contexts containing the seed facts, which are clustered to generate new patterns. The patterns are given confidence scores, and the best performing ones are used to mine for new facts to retrain the system. Agichten et al. [1] improved Snowball by using sparse Markov transducers to represent contexts, enabling the coding of word order. StatSnowball [26] also built on Snowball but used Markov logic networks, supplying a measure of pattern confidence in a more natural way. KnowItAll [11] and LEILA [19] also applied bootstrapping, the latter incorporating both positive and negative examples as seeds.

Banko et al. [10] introduced Open Information Extraction, a scenario where the relations are not known in advance and the corpora involved are massive and heterogeneous. Textrunner, the first such system [10] uses a Naive Bayes classifier to predict whether tokens between two entities indicate a relationship or not. The authors continued this work [4] with a system that uses conditional random fields (CRF), trained by self-supervision: a small number of relation-independent heuristics are applied to generate labelled (positive and negative) examples.

Using Wikipedia. A number of projects have exploited Wikipedia for information extraction. DBpedia [3] and Yago [22] extract information from the structured parts (infoboxes, categories, lists etc.) Suchanek et al. [22] construct an ontology by mapping the extracted entities to WordNet [12].

Ruiz-casado et al. [17] proposed a method for automatic extraction and generalization of extraction patterns for semantic relationships (hyperonymy/hyponymy, holonymy/meronymy) from Simple English Wikipedia, using and extending WordNet. The extraction patterns are generalized with an algorithm using minimum edit distance, using a representation resembling the one proposed by this paper (see Section 5).

Culotta et al. [8] present a model to integrate information extraction and data mining, demonstrated on Wikipedia articles. They apply CRFs, using both contextual and relational features. Nguyen et al. [14] demonstrate relation extraction from Wikipedia article free texts using dependency tree mining and supervised machine learning with SVM classifiers. Similar to our system, they use a custom coreference resolution algorithm exploiting special characteristics of Wikipedia pages. They also use a custom named-entity type recognition relying on supervised classification of wikipedia pages corresponding to the entities in the relations.

Suchanek et al. [21] present PORE, an algorithm for situations involving only positive and unlabeled examples, applied to semi-automatic IE from free text in Wikipedia articles. The algorithm is based on an SVM classifier, and uses bootstrapping, strong negative identification and transductive inference. Just as in our approach, positive training examples are generated from infobox data. Entities are characterized by features from their respective pages.

The Kylin/KOG project [23] is a complex Open IE system that uses Wikipedia infoboxes to generate training data for document and sentence classifiers and CRF relation extractors that are run on the free text sections of the articles. Training data sparseness problem is solved by generating an ontology of Wikipedia infobox

schemata and using the inheritance of relations. Training data is also augmented by Google searches for the Wikipedia page titles and finding sentences containing the infobox data. Extraction is also extended to web search results, increasing recall and precision.

The SOFIE system [20] presents an integrated approach involving 1st-order predicate logic representation and logical reasoning to solve pattern extraction, entity disambiguation and consistency checking together in one unified model. The system relies on knowledge in the Yago ontology and is able to extend it with information extracted from Wikipedia and web searches with high precision.

Weld et al. [24] propose a system for unsupervised relation extraction, the task of automatically discovering interesting relations between entities. Sentences in Wikipedia pages containing candidate entities (anchor texts linked to other Wikipedia pages) are clustered in two steps, using features from dependency parsing for high precision and additional surface patterns obtained from web searches for increasing coverage.

Using deep NLP methods. Several papers have proposed to use features from deep linguistic processing (parsing) for information extraction. Yan et al. [25] do relation extraction using a tree kernel defined over shallow parse tree representations of sentences. Culotta et al. [9] continue this work by defining a tree kernel over relation instances consisting of the smallest dependency tree containing the two entities of the relation. Bunescu et al. [7] improve this by using the shortest path between the entities in the dependency graph, while Nguyen et al. [14] use dependency subtrees. Yan et al [24] use the subpaths in the shortest path connecting the two entities in the dependency trees as features for their clustering algorithm.

Our system also uses a deep parser to leverage syntactic structures from the input text, however, it differs from these approaches in the respect that instead of dependency trees, our system uses phrase structures and verb frames derived from these (see Section 3).

3 Information Extraction with Verb Frames

The domain for the development of our baseline system was the *studies* relation, where a person's Wikipedia article is searched for fillers for the following slots: the name of the educational institution where the person studied; starting and ending times of studies; time of obtaining qualification; name of obtained qualification (i. e. type of degree: B.A., M.A., Ph.D etc.), and field of study. For example, the sentence

> In 1977, he graduated magna cum laude from Harvard University with a B.A. in mathematics.

would produce the following fillers for the studies template:

```
School name: Harvard University
Begin studies date: -
End studies date: -
Qualification date: 1977
Qualification: B.A.
Field(s): mathematics
```

3.1 Corpus Construction

The base of our Wikipedia corpus was the June 2008 version of the static dump of English Wikipedia pages[3], containing about 2.4 million articles. We used simple heuristics, such as checking for information about birth and death dates, etc. in order to identify about 100,000 autobiographical articles with high accuracy. These articles were processed to separate raw text content (containing only paragraph boundary information) from formatting and other page elements. Meta-information, such as page title and title variants (obtained by processing redirection page links), category labels, hyperlinks within the text etc. were retained in separate files to facilitate later processing.

3.2 Linguistic Analysis and Pattern Matching

The raw text paragraphs were processed by LingPipe's sentence segmentation tool[4], followed by parsing with Enju [18], an efficient and wide-coverage English parser using a probabilistic Head-Driven Phrase Structure (HPSG) grammar. Enju is capable of producing both phrase structures and predicate-argument structures. We identified the verb frame structures in the parses that would be matched by the IE patterns. The noun phrases in the verb frames were processed by special named entity recognizers. We will now describe these two steps in more detail.

In the parser's output, we first identified verb phrases (clauses) inside the sentence that carried relevant information: coordinate clauses, relative clauses, some prepositional phrases with a verb phrase (VP) complement having "before" or "after" for prepositions etc. We skipped VPs having negated or non-declarative main verbs.

Special care had to be taken when processing the noun phrases. The top-level NPs in Enjus grammar can have complex internal structures covering many terminals, of which not all would be necessary for information extraction (e.g. a PP at the end, as in "Juilliard School in New York City".) We therefore selected terminals in the noun phrases up to and including the head, plus any following tokens participating in structures analyzed as apposition or possession, both of which could be legitimate parts of the names (e.g. "Montana School of Mines", "December, 1988".) Determiners, possessive pronouns, prepositions etc. were removed from the front of the NPs. For each selected terminal in the noun phrases, we recorded its surface form, base form, part-of-speech tag and sentence position.

[3] http://static.wikipedia.org/downloads/2008-06/

[4] http://alias-i.com/lingpipe/

Coordinated phrases were split and we produced all possible combinations with the other sentence constituents. For example, the following complex sentence,

> After receiving a Bachelor's Degree in mathematics and physics at the University of Michigan, he went on to obtain a Ph.D. in electrical engineering at Harvard in 1998.

would produce the following structures after parsing and processing:

```
((Verb, "receive"), (Subj, "he"),
 (Obj, "Bachelor's Degree"), (PP-in, "mathematics"))
((Verb, "receive"), (Subj, "he"),
 (Obj, "Bachelor's Degree"), (PP-in, "physics"))
((Verb, "go on"), (Subj, "he"), (Verb2, "obtain"),
 (Obj2, "Ph.D."),  (PP-in2, "electrical engineering"),
 (PP-at2, "Harvard"), (PP-in2, "1998"))
```

The recognition of frame slots (such as name of school, field of study etc.) is based on syntactic and semantic constraints. The syntactic constraints match the grammatical role of the given NP in the sentence. The semantic constraints ensure that only the right types of entities are matched. For instance, an extraction pattern would look like the following (in simplified form):

```
Subj(PERSON)+V('attain')+Obj(DEGREE)+PP-in(SCHOOL)+PP-in(DATE)
```

This means that the main verb must be (a form of) "attain", the subject NP must be of type PERSON, the object NP must be of type DEGREE, and prepositional phrases headed by "in" either designate the SCHOOL or the DATE slots of the relation. To check the semantic constraints, we applied simple, custom-made named-entity recognition using regular expressions and/or lexicons that were used to check the heads of the NPs. We had to employ custom NER solutions because most freely available NER-taggers can only recognize standard general categories such as *person, location, organization* etc. which are insufficient for the *studies* relation. We used several online sources, dictionaries, thesauri etc. to compile the lexicons as extensively as possible. The fields of study lexicon, for example, contains about 2,100 entries, while the educational institution names lexicon comprises more than 34,000 items.

In order to ensure that we were extracting information about the person in focus, we checked for references to the article title person in the input text. We checked for occurrences of the page title, its name variants obtained from the redirection links, its substrings (to account for further name variations), or personal pronouns.

The baseline system used about 20 extraction patterns that were constructed by human knowledge experts after several person days of time was spent on studying a sample of articles in the domain. We created a human-annotated development corpus in order to aid the construction of the baseline system. 200 "person" articles were randomly chosen and the relevant information slots were manually tagged by

2 annotators, while a 3rd annotator checked the results. We periodically performed automatic tests against this gold standard during development. As a design principle, the system was optimized for precision in the precision-recall tradeoff, since reliability was declared to be crucial, while a desired recall of at least 40% would still yield considerable amount of data given Wikipedia's vast coverage.

For the evaluation of the final baseline system, further 100 randomly chosen Wikipedia person articles were prepared by the human annotators. We calculated precision and recall of recognition of frame slots against this set (see Table 1).

Table 1: Evaluation of information extraction with manually developed patterns in the "studies" domain

Precision	Recall	F-measure
94.22%	60.33%	73.56%

3.3 Special Problems

During the development of the baseline system, we encountered several problems, which involved finding workarounds for several re-occurring errors in the output of Enju parser.

The first is the well-known prepositional phrase-attachment problem, when the parser attached the same type of PPs inconsistently to either the main VP of the clause or the final NP preceding the PP. For this reason, we ignored the structural relations proposed by the parser for the PPs, and used our own heuristics in order to identify the required dependencies (for example, in the case of time adverb PPs, rules based on relative sentential order.)

A second, very common problem was presented by the parser's failure to correctly identify the boundaries of multi-word proper names and other named entities, resulting in incorrect parses. To overcome this, we tried to recognize as many named entities as possible before parsing, using special characters to merge them into single tokens. These would be treated by Enju as single-word nouns in the input sentence thus producing the correct syntactic analyses. We used several simple but reliable methods that would not require the complex resources of a dedicated named-entity recognizer run on tens of thousands of documents. One such method was to search the original raw text for hyper-links (referring to other Wikipedia pages). If anchor texts within such links contained multi-word proper names, these were marked, as they would refer to proper name entities with a high probability. We also generated a list of potential names from all the multi-word capitalized page titles of all the Wikipedia pages in our static dump, and tried to recognize and mark these in the input articles.

Another, similar problem occurred when the parser incorrectly analyzed several common named entity types containing commas, such as dates ("April, 1996"), or school names ("University of Berkeley, California") as coordinated NPs. We applied

recognition and marking before parsing for these categories as well. The dates were recognized by regulars expressions, while the recognition of school names used a combination of regular expressions and special lexicons, using the above-mentioned educational institution names directory in conjunction with 2.3 million geographical names.

4 Information Extraction Using Machine Learning

We conducted experiments with supervised machine learning methods, depending on shallow, less resource-intensive linguistic analysis, and compared it to the baseline method. The domain was the same as the baseline system's (*studies*). Training instances were generated automatically, by looking up instances of the Wikipedia "alumni" category labels, entries of the academic degree names lexicon described above, and simple regular expressions for dates in the articles' texts. These instances were then hand-checked, yielding a corpus of about 2000 annotated training examples. However, this method provided us with annotations for only 3 of the 6 slots in the *studies* scenario (name of educational institution, qualification name, year of qualification), since there was no redundant Wikipedia information available to automatically identify the other entities in the original relation presented in Section 3.

The training documents were only processed by sentence segmentation, tokenization and part-of-speech tagging. We trained the Mallet maximum entropy classifier[5]. The learning features were n-grams (n=1,2,3 before and n=1,2 after) and the base forms of the nearest verbs in the sentence.

We used the human-annotated evaluation set described in Section 3 to evaluate the performance of the classifier and to compare it with the baseline method that used deep parsing, special named entity recognizers and hand-crafted extraction patterns. We measured precision and recall for each of the 3 slots that were extracted. We were also interested in how the two methods complemented each other, so we also evaluated the union and the intersection of the results coming from the two systems (Table 2).

The machine learning approach came closest in precision to the pattern-based approach in the recognition of institution and academic degree names, while recall was significantly lower for both. The intersection of the two methods yielded 100% precision at a cost of very low recall. The union of the two methods, however, in the case of institution and degree names did not degrade the precision of the machine learning approach, but brought a significant increase in recall even in comparison to the better-performing pattern-based approach. This suggests that the two approaches tend to complement each other, each working well on different types of instances. The resulting hybrid system could be used well for practical applications, since its precision remained above the critical 90% threshold while its recall was improved.

[5] `http://mallet.cs.umass.edu`

Table 2: Comparison of pattern-based (PB) and machine learning (ML) methods for 3 slots of the "studies" frame (precision, recall, F-measure) (decimals were retained to conserve space)

	Institution			*Date*			*Qualification*		
	P	R	F1	P	R	F1	P	R	F1
PB	92%	67%	78%	100%	55%	71%	94%	63%	75%
ML	91%	41%	57%	85%	47%	61%	92%	44%	60%
Union	91%	76%	83%	90%	75%	82%	94%	81%	87%
Intersection	100%	11	20%	100%	5%	10%	100%	2%	4%

The lower precision of qualification dates in the machine learning results could be explained by the fact that it had no knowledge that dates could participate in three different roles in these contexts (beginning and ending time of studies, time of obtaining qualification), in contrast to the baseline system.

5 Learning Extraction Patterns

We intended to develop a system that would be able to learn extraction patterns automatically from pre-annotated example sentences. Our goal was to create a general framework that could be adapted to new extraction domains in the shortest time possible, utilizing minimal human effort. A human knowledge expert would only be required to check, and if necessary, edit the extraction patterns that were automatically generated by the system. The annotator could also decide to mark certain patterns as negative, indicating constructions that are similar to real patterns, but produce incorrect results (for example, in the above-mentioned *studies* domain, one would want to exclude sentences about receiving honorary academic degrees, as opposed to real academic degrees.)

5.1 Generating Training Instances

In order to reduce the effort of creating annotated training instances, we relied on the assumption that a certain degree of redundancy could be expected between the structured and free text contents of Wikipedia. We used the results of the Yago project [22] in order to gain access to the information in the structured sections (tables, category labels) of the English Wikipedia pages. Part of Yago's knowledge is available in the form of binary relations between Wikipedia entities (Wikipedia entries.) We harvested free-text training examples automatically by looking up sentences contained in the articles about the 1st arguments that contained references to the 2nd arguments of the given Yago relations.

We used the *awards* relation for the development of the extraction pattern learning system, which holds between two entities (awarded person, award name).

A baseline system with manually developed extraction patterns, similar to the one described in Section 3 had been also developed for this domain and was available for comparison.

We used Yago's *hasWonPrize* relation, which holds between *person* and *award* Wikipedia entities. Looking up the award names in the persons' pages produced about 16,000 potential training sentences. Since the *hasWonPrize* relation was noisy not only award names, but titles of award-winning works like film and album titles were present as 2nd arguments, we used a lexicon of about 7,400 award names to filter out the misleading sentences, leaving about 13,000 for further processing.

The sentences were processed by Enju parser and the verb frame extraction process described in Section 3. We identified and annotated the two Yago arguments inside the identified constituent noun phrases. In order to recognize the 1st argument (person name), we applied simple rule-based coreference resolution. We looked for the page title, its variant extracted from the 1st sentence in the leading paragraph, or any token-substrings of these to cover other name variations. If none of these could be recognized under any NP, we looked for personal pronouns, taking into account the gender of the person corresponding to the page title. We counted the number of feminine and masculine 3rd person singular personal pronouns and assumed the gender of the title person to correspond with the gender having the higher count.

After the annotation of the Yago arguments, we excluded sentences that did not contain entities for both slots, leaving about 11,000 training sentences.

In order to facilitate the generalization of extraction patterns from the training sentences, we also annotated several named entity types that could be recognized easily using regular expressions (ordinals, cardinals, various date formats, month names, years, and numbers.)

5.2 Generating Extraction Patterns

Our goal was to generate extraction patterns from the annotated training sentences taking the following two criteria into account: 1) the number of generated patterns should be as low as possible to support human post-processing, but at the same time these patterns should cover as many as possible of the original training sentences, 2) the generated patterns should have a uniform syntax, and should be easy to read and edit by humans. Manual editing should mainly constitute deleting unapproved patterns or pattens elements.

The outline of our proposed pattern learning algorithm is the following:

1. Converting training instances to patterns
2. Creating pattern classes
3. Identification of marker tokens
4. Creating generalized patterns from the pattern classes

In the following, we describe each step in detail together with the results of our experiment with its application in the *awards* domain.

1. Converting training instances to patterns. All of the training sentences, annotated with syntactic constituents, Yago relation arguments and simple named entity categories were converted to patterns. A pattern is a list of ordered pairs *(G, S)*, where G is the name of a grammatical role (*Verb, Subj, Obj, ObjII* and *PP-xx*, *xx* being an English preposition), while S is a list of tokens inside the constituent having label G. S may consist of either meta-tokens (Yago argument or named entity labels), or simple sentence tokens (see examples below.)

2. Creating pattern classes. In the next step, we merged identical patterns that were found in different sentences, keeping pointers to the original containing sentences for later reference. Then we grouped the patterns into classes, assuming two patterns to be in the same class if: 1) both patterns had the same lexical value for the Verb constituents, 2) the two Yago arguments were located under the same constituent labels.

As a result, the original 11,000 training sentences were grouped into 376 different pattern classes. The classes were ranked according to the total number of training sentences the class members covered. We found that there were only 64 classes that contained at least 2 patterns, and that these covered about 97% of all the original training sentences.

In the following example, we show pattern classes ranked #1 and #4 and a few of their pattern elements along with the number of training sentences covered by each:

```
Class id: 8
Sentences covered by patterns in class: 1092
Patterns in class: 210
(('Verb', 'win'),
 ('Subj', '#PERSON#'), ('Obj', '#PRIZE#'))     548
(('Verb', 'win'),
 ('Subj', '#PERSON#'), ('Obj', '@CARDINAL@ #PRIZE#s')) 99
(('Verb', 'win'),
 ('Subj', '#PERSON#'), ('Obj', '@YEAR@ #PRIZE#'))      98
(('Verb', 'win'),
 ('Subj', '#PERSON#'), ('Obj', '@ORDINAL@ #PRIZE#'))   48
(('Verb', 'win'),
 ('Subj', '#PERSON#'), ('Obj', 'Daytime #PRIZE#'))     22
...

Class id: 5
Sentences covered by patterns in class: 406
Patterns in class: 258
(('Verb', 'be'),
 ('Subj', '#PERSON#'), ('Obj', '#PRIZE#'))      27
(('Verb', 'be'),
```

```
 ('Subj', '#PERSON#'),
 ('Obj', '#PRIZE# -winning American actor'))      18
(('Verb', 'be'),
 ('Subj', '#PERSON#'),
 ('Obj', '#PRIZE# -winning American actress'))    18
(('Verb', 'be'),
 ('Subj', '#PERSON#'),
 ('Obj', 'recipient of the #PRIZE#'))     10
(('Verb', 'be'),
 ('Subj', '#PERSON#'),
 ('Obj', 'American #PRIZE# -winning actor'))      8
...
```

In this example, pattern class with id "5" contains 258 patterns, covering 406 training sentences. All the patterns in this class have "be" as the main verb, the 1st Yago argument (person name) is in the subject position, while the 2nd argument (award name) is in the object position.

3. Identification of marker tokens. As it can be seen in the example patterns above, the pattern constituents may contain a number of different tokens around the award name (marked by the #PRIZE# meta-token.) Some of these tokens obviously indicate that the sentence corresponds to the event in question (e.g. "winner", "recipient" etc.), while others obviously not ("actor", "actress" etc.) Our goal was to attempt to automatically identify marker tokens that correlate with the extraction domain, and to set them apart from non-markers that are irrelevant and thus could be omitted from the suggested generalized patterns.

For the identification of marker tokens, we used Pearson's one-sided χ^2-test [15]. To achieve this, we needed negative examples, which, in contrast to the positive sentences described in the previous section, were not related to the relation under examination. These were produced by taking all the sentences in the persons' articles that did not include any of the positive sentences. Since these outnumbered the positives, we took a random sample to balance the two categories. Using the χ^2-test, we selected those tokens that did not show independence of the two categories. We used an empirically set threshold of 25.0 for the test (the critical value would have been 6.635 for $\alpha = 0.01$, but we chose a more strict threshold after some empirical tests.)

4. Creating generalized patterns from the classes. In the last step, we generated a single suggested general pattern for each pattern class, containing only the marker words identified by the χ^2-test. These suggested patterns were then given to a human knowledge expert for reviewing and categorizing into positive or negative patterns. The following generalized patterns were produced for the two pattern classes shown above:

```
Class id: 8
+(('Verb', 'win'),
 ('Subj', '* #PERSON# *'), ('Obj', '* #PRIZE# *'), ('ObjII', '*'))
```

```
-(('Verb', 'win'),
 ('Subj', '* #PERSON# *'),
 ('Obj', '* #PRIZE# -nomination|nomination'), ('ObjII', '*'))

Class id: 5
+(('Verb', 'be'),
 ('Subj', '* #PERSON# *'),
 ('Obj', '* #PRIZE# -winning|recipient|winner|winning'))
+(('Verb', 'be'),
 ('Subj', '* #PERSON# *'), ('Obj', 'recipient|winner #PRIZE# *'))
```

The *S* component of the *(G, S)* ordered pairs in the generalized patterns may contain disjunctive lists of tokens, separated by the "|" character, meaning that at least one of these items must be present in the given position for the pattern to match. The special character "*" means that any token can stand in that position. The "+" prefix indicates patterns marked positive, and the "-" prefix indicates patterns marked negative by the annotator (in the above example, sentences about award nominations are excluded by the patterns.)

5.3 Evaluation

In the *awards* domain, a human domain expert reviewed 43 suggested patterns that belonged to pattern classes covering at least 5 positive training sentences, covering 92% of all the positive training sentences. The work took about 1.5 hours, and produced 28 positive and 5 negative patterns.

Information extraction with the automatically extracted and human-approved patterns used the same parsing and named entity recognition methods that were described in Section 5.1. The annotated input text was first checked for negative patterns, then the remaining sentences were matched against the positive patterns.

We evaluated the results against a manually annotated corpus of 100 randomly selected person articles. Table 3 shows the precision and recall results of the evaluation of information extraction using both the automatically extracted patterns and the baseline system using completely manually developed extraction patterns.

Table 3: Evaluation of information extraction using automatically extracted (AE) and manually developed (PB) patterns in the "awards" domain

	Precision	Recall	F-measure
AE	91.66%	36.70%	52.41%
PB	93.97%	50.00%	65.27%

It can be seen from Table 3 that the precision of the automatically extracted patterns comes close to the manual system's. Recall, however, is significantly higher in the case of manually created patterns. It could likely be improved by adding more

lower-ranked suggested patterns for human revision, which would require more time for revision but would exploit more information from the training sentences.

We were interested in how the pattern extraction framework would fare in other domains, with different types of relations and entities. We used Yago's *IsMarriedTo* relation, which holds between *person* entities (we will refer to this as the *spouse* domain), to generate a training corpus of about 1000 sentences, and used the framework to derive suggestions for extraction patterns. 31 patterns (25 positive and 6 negative) were approved by the knowledge expert in the end, requiring about 2 hours of work.

Preliminary tests showed that this domain would present challenges to a regular expression- and lexicon-based name recognizer, therefore we added the dedicated Stanford named-entity recognizer [13] to aid the correct recognition of person name boundaries in the input sentences. In addition, we also used simple, rule-based coreference-resolution [5] in order to track mentions of the proper names throughout the text and to be able to produce normalized forms of names in the output.

The system adapted to this domain was used to extract marriage relations inside 100 persons' Wikipedia pages, which were previously human-annotated with the correct answers. We were also curious about how it would compare to a machine-learning solution for this domain, so we trained the maximum entropy classifier on the 1000-sentence training set described above with the features described in Section 4. As before, we also carried out evaluation of the union and intersection combinations of the two methods to see how they complemented each other. The evaluation results can be seen in Table 4.

Table 4: Evaluation of information extraction using automatically extracted patterns (AE) and machine learning (ML) in the "spouse" domain

	Precision	Recall	F-measure
AE	89.97%	35.30%	50.71%
ML	90.43%	53.19%	66.98%
Union	90.36%	61.87%	73.45%
Intersection	100.00%	22.26%	36.41%

Machine learning outperformed the pattern-based approach in terms of recall, while the precision of the two approaches was nearly identical. A possible explanation could be hypothesized from the fact that Stanford NER's more general classifier is outperformed by the dedicated classifier that was trained on data that is more similar to the evaluation data (person names inside Wikipedia articles), leading to a higher coverage of recognized person names and thus a higher recall in the relation extraction. The combination tests revealed in this case, too, that the two approaches are likely to miss out in different situations. The union of the two result sets produced the same precision but a significantly higher recall than than either of the two methods.

6 Discussion and Conclusion

We have described several methods for the reliable extraction of massive numbers of facts from the Wikipedia online encyclopedia. In practice, for the *studies*, *awards* and *spouse* domains, we successfully applied these methods to extract more than 70,000 facts for about 35,000 persons in Wikipedia. In addition, we also created and applied solutions to learn properties of *award entities* (name of award, date it was first awarded, who is it awarded by, who are awarded, who was the award named after) and *geographical regions* (name of region, capital, containing and contained regions) using Wikipedia. This information was integrated into the semantically linked database of the iGlue project[6].

We have described a frame-based information extraction system that relies on deep natural language processing and manually crafted extraction patterns. This approach can be useful if no annotated training data is available, or when producing such annotations could be too costly (for example, when there are too many different slots in the extraction frame, as it is the case with the *studies* relation which has 6 slots (see Section 3)).

In our pattern-based approaches, the use of reliable, custom-made named entity recognizers is crucial. However, in many cases, this task can be overcome by resource-friendly methods such as regular expressions and/or lexicons, which can be complemented by simple but effective heuristics that utilize the special advantages of Wikipedia (hyperlinks, page titles, tables and category labels, special formatting etc.)

We have also proposed an approach for generating extraction patterns from labeled data, requiring only minimal amount of work on behalf of human knowledge experts. We have demonstrated that much of the burden of preparing training data can be reduced by utilizing existing semantic resources, such as the Yago Ontology, and taking advantage of both the redundancy and the volume of information found in an encyclopedia such as Wikipedia. The performance of information extraction using automatically generated patterns and the fraction of the human effort can be well compared to completely manually created systems, when precision is of prime importance.

We also compared the performance of the pattern-based approaches, using both manually and automatically generated patterns to the performance of machine learning solutions using state-of-the-art supervised algorithms and less resource-intensive natural language processing procedures. Our results indicated that while precision is comparable for the approaches, recall was favored by different approaches. In more detail, the experiments revealed that in terms of recall, manual patterns outperform machine learning (Table 2) and automatically extracted pattens (Table 3), while machine learning has better recall than automatically extracted patterns (Table 4). This suggests an order in the (F-measure) performance of the three approaches:

manual patterns (PB) > machine learning (ML) > automatically extracted

[6] `http:\\iglue.com`

patterns (AE).

While the superiority of the manually constructed patterns over the two other approaches is more obvious, the relationship between the performance of the machine learning and the pattern extraction methods raises new questions. In particular, it should initiate new experiments in order to tune the performance of our pattern learning algorithm.

The examination of the combination of the machine-learning and pattern-based approaches (using both manually and automatically generated patterns) revealed that they complement each other well, leading to a straightforward way of extending the performance of the pattern-based system. In the future it would be interesting to experiment with a deeper synergy of the two methods, for example using more sophisticated features available from deep parsing to train the classifiers.

References

[1] Agichtein, E., Eskin, E., and Gravano, L. Combining strategies for extracting relations from text collections. In *Proceedings of the 2000 ACM SIGMOD Workshop on Data Mining and Knowledge Discovery*, 2000.

[2] Agichtein, Eugene and Gravano, Luis. Snowball: Extracting relations from large plain-text collections. In *In Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94, 2000.

[3] Auer, Sören, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, Cyganiak, Richard, and Ives, Zachary. Dbpedia: a nucleus for a web of open data. In *ISWC'07/ASWC'07: Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[4] Banko, Michele and Etzioni, Oren. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[5] Bontcheva, Kalina, Dimitrov, Marin, Maynard, Diana, Tablan, Valentin, and Cunningham, Hamish. Shallow methods for named entity coreference resolution. In *TALN 2002*, June 2002.

[6] Brin, Sergey. Extracting patterns and relations from the world wide web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK, 1999. Springer-Verlag.

[7] Bunescu, Razvan C. and Mooney, Raymond J. A shortest path dependency kernel for relation extraction. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[8] Culotta, Aron, McCallum, Andrew, and Betz, Jonathan. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[9] Culotta, Aron and Sorensen, Jeffrey. Dependency tree kernels for relation extraction. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[10] Etzioni, Oren, Banko, Michele, Soderland, Stephen, and Weld, Daniel S. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, December 2008.

[11] Etzioni, Oren, Cafarella, Michael, Downey, Doug, Kok, Stanley, Popescu, Ana-Maria, Shaked, Tal, Soderland, Stephen, Weld, Daniel S., and Yates, Alexander. Web-scale information extraction in knowitall. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM.

[12] Fellbaum, C., editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[13] Finkel, Jenny R., Grenager, Trond, and Manning, Christopher. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[14] Nguyen, Dat P. T., Matsuo, Yutaka, and Ishizuka, Mitsuru. Relation extraction from wikipedia using subtree mining. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 1414–1420. AAAI Press, 2007.

[15] Plackett, R. L. Karl pearson and the chi-squared test. *International Statistical Review / Revue Internationale de Statistique*, 51(1):59–72, 1983.

[16] Riloff, Ellen. Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*, pages 1044–1049, 1996.

[17] Ruiz-casado, Maria, Alfonseca, Enrique, and Castells, Pablo. Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In *Proceedings of the Atlantic Web Intelligence Conference, AWIC-2005. Volume 3528 of Lecture Notes in Computer Science*, pages 380–386. Springer Verlag, 2005.

[18] Sagae, Kenji and Miyao, Yusuke. Hpsg parsing with shallow dependency constraints. In *In Proc. ACL 2007*, 2007.

[19] Stevenson, M. and Greenwood, M. A. Dependency pattern models for information extraction. *Research on Language and Computation*, 7(1):13–39, 2009.

[20] Suchanek, Fabian M., Ifrim, Georgiana, and Weikum, Gerhard. Combining linguistic and statistical analysis to extract relations from web documents. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717, New York, NY, USA, 2006. ACM Press.

[21] Suchanek, Fabian M., Kasneci, Gjergji, and Weikum, Gerhard. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, 2008.

[22] Suchanek, Fabian M., Sozio, Mauro, and Weikum, Gerhard. Sofie: a self-organizing framework for information extraction. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 631–640, New York, NY, USA, 2009. ACM.

[23] Wang, Gang, Yu, Yong, and Zhu, Haiping. Pore: positive-only relation extraction from wikipedia text. In *ISWC'07/ASWC'07: Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 580–594, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] Weld, Daniel S., Hoffmann, Raphael, and Wu, Fei. Using wikipedia to bootstrap open information extraction. *SIGMOD Rec.*, 37(4):62–68, 2008.

[25] Yan, Yulan, Okazaki, Naoaki, Matsuo, Yutaka, Yang, Zhenglu, and Ishizuka, Mitsuru. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1021–1029, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[26] Zelenko, Dmitry, Aone, Chinatsu, and Richardella, Anthony. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, 2003.