Extracting Semantic Concept Relations from Wikipedia

Patrick Arnold
Inst. of Computer Science, Leipzig University
Augustusplatz 10
04109 Leipzig, Germany
arnold@informatik.uni-leipzig.de

Erhard Rahm
Inst. of Computer Science, Leipzig University
Augustusplatz 10
04109 Leipzig, Germany
rahm@informatik.uni-leipzig.de

ABSTRACT

Background knowledge as provided by repositories such as WordNet is of critical importance for linking or mapping ontologies and related tasks. Since current repositories are quite limited in their scope and currentness, we investigate how to automatically build up improved repositories by extracting semantic relations (e.g., is-a and part-of relations) from Wikipedia articles. Our approach uses a comprehensive set of semantic patterns, finite state machines and NLP-techniques to process Wikipedia definitions and to identify semantic relations between concepts. Our approach is able to extract multiple relations from a single Wikipedia article. An evaluation for different domains shows the high quality and effectiveness of the proposed approach.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Languages

1. INTRODUCTION

Background knowledge plays an important part in information integration, especially in ontology matching and mapping, aiming at finding semantic correspondences between concepts of related ontologies. There are numerous tools and approaches for matching ontologies that mostly focus on finding pairs of semantically equivalent concepts [22, 3, 21, 4]. Most approaches apply a combination of techniques to determine the lexical and structural similarity of ontology concepts or to consider the similarity of associated instance data. The lexical or string similarity of concept names is usually the most important criterion. Unfortunately, in many cases the lexical similarity of concept names does not correlate with the semantic concept similarity due to uncoordinated ontology development and the high complexity of language. For example, the concept pair (car, automobile) is semantically matching but has no lexical similarity, while there is the opposite situation for the pair (table, stable). Hence, background knowledge sources such as synonym tables and dictionaries are frequently used and vital for ontology matching.

The dependency on background knowledge is even higher for *semantic ontology matching* where the goal is to identify not only pairs of equivalent ontology concepts, but all related concepts together with their semantic relation type, such as is-a or part-of. Determining semantic relations obviously results in more expressive mappings that are an important prerequisite for advanced mapping tasks such as ontology merging [23, 24] or to deal with ontology evolution [13, 9]. Table 1 lists the main kinds of semantic relations together with examples and the corresponding linguistic constructs. The sample concept names show no lexical similarity so that identifying the semantic relation type has to rely on background knowledge such as thesauri.

Relatively few tools are able to determine semantic ontology mappings, e.g., S-Match [8], TaxoMap [12] as well as our own approach [1]. All these tools depend on background knowledge and currently use WordNet as the main resource. Our approach [1] uses a conventional match result and determines the semantic relation type of correspondences in a separate enrichment step. We determine the semantic relation type with the help of linguistic strategies (e.g., for compounds such as 'personal computer' is-a 'computer') as well as background knowledge from the repositories WordNet (English language), OpenThesaurus (German language) and parts of the UMLS (medical domain). Together with the match tool COMA [16] for determining the initial mapping, we could achieve mostly good results in determining the semantic relation type of correspondences. Still, in some mapping scenarios recall was limited since the available repositories, including WordNet, did not cover the respective concepts.

Based on the previous evaluation results, we see a strong need to complement existing thesauri and dictionaries by more comprehensive repositories for concepts of different domains with their semantic relations. To build up such a repository automatically, we aim at extracting semantic correspondences from Wikipedia which is the most comprehensive and up-to-date knowledge resource today. It contains almost any common noun of the English language, and thus presumably most concept names. Articles are user-generated and thus of very good quality in general. Furthermore, Wikipedia content can be accessed free of charge.

WIMS '14, June 2-4, 2014 Thessaloniki, Greece

Table 1: Semantic concept relations.

Relation type	Example	Linguistic relation
equal	river, stream	Synonyms
is-a	car, vehicle	Hyponyms
has-a	body, leg	Holonyms
part-of	roof, building	Meronyms

In this paper we make the following contributions:

- We present a novel approach to extract semantic concept correspondences from Wikipedia articles. We propose the use of finite state machines (FSM) to parse Wikipedia definitions and extract the relevant concepts.
- We use a comprehensive set of *semantic patterns* to identify all kinds of semantic relations listed in Table 1. The proposed approach is highly flexible and extensible. It can also extract multiple relations from a single Wikipedia article.
- We evaluate our approach against different subsets of Wikipedia covering different domains. The results show the high effectiveness of the proposed approach to determine semantic concept relations.

In the next section we discuss related work. Section 3 introduces the notion of semantic patterns and outlines which kinds of patterns we use for discovering semantic relations. Section 4 describes the new approach to extract semantic relations from Wikipedia in detail. In Section 5 we evaluate the approach for different test cases. We conclude with a summary and outlook (Section 6).

2. RELATED WORK

Background knowledge about concepts and named entities with their relationships is maintained in numerous repositories (e.g., thesauri) that are either manually, semi-automatically or fully automatically developed and maintained. One of the oldest and most popular repositories is WordNet[1] which has its roots in the mid-1980s [17]. Its content is manually derived by linguists, making it a highly precise resource. However, progress is relatively slow and WordNet lacks many modern terms, e.g., *netbook* or *cloud computing*.

Crowd sourcing is a promising approach to speed-up the laborious development of a comprehensive thesaurus by utilizing a community of volunteers. An exemplary effort is OpenThesaurus (German language thesaurus). As the contributors are no linguistic experts, we discovered that the precision is slightly below WordNet, though.

The development of large knowledge repositories with some millions of elements and relationships is only feasible with automatic approaches for knowledge acquisition from existing text corpora and especially from the web. This can either be done by directly extracting knowledge from documents and web content (e.g., Wikipedia) or by exploiting existing services such as web search engines. The latter approach is followed in [11], where a search engine is used to check the semantic relationship between two terms A and B. They send different phrases like "A is a B" (like "a computer is a device") or "A, such as B" (like "rodents, such as mice") to a search engine and decide about the semantic relation based on the number of returned search results. Such an approach is typically not scalable enough to build up a repository since the search queries are rather time-consuming and since there are typically restrictions in the allowed number of search queries. However, such approaches are valuable for verifying found semantic correspondences, e.g., for inclusion in a repository or for ontology mapping.

Numerous research efforts aimed at extracting knowledge from Wikipedia, as a comprehensive and high quality (but textual) web information source and lexicon. The focus and goals of such efforts vary to a large degree. Examples include approaches that extract generalized collocations [5], computing semantic relatedness between concepts or expressions [6], [28] and word sense disambiguation [19]. More related to our work are previous efforts to derive structured knowledge and ontologies from Wikipedia, for example DBpedia, Yago, Freebase and BabelNet. *DBpedia* [2] focusses on the extraction of structured content from info boxes in Wikipedia articles which is generally easier than extracting content from unstructured text. The extracted knowledge is mostly limited to named entities with proper names, such as cities, persons, species, movies, organizations etc. The relations between such entities are more specific (e.g., "was born in", "lives in", "was director of" etc.) than the general relation types between concepts that are more relevant for ontology mappings and the focus of our work.

The *Yago* ontology [29] enriches DBpedia by classifying Wikipedia articles in a thesaurus, as the Wikipedia-internal categories are often quite fuzzy and irregular. Yago thus contains both relations between entities, e.g., "Einstein was a physicist", as well as linguistic/semantic relations, e.g., "physicist is a scientist". The latter relations are derived by linking Wikipedia articles from category pages to the WordNet thesaurus. We experimented with Yago, but found that it is of relatively little help if WordNet is already used, e.g., Yago will not link concepts A and B if neither is contained in WordNet.

BabelNet is a popular NLP project focusing more strongly on linguistic relations [18]. It contains millions of concepts and relations in multiple languages and utilizes mappings between Wikipedia pages and WordNet concepts. Its precision is around 70-80 %, depending on the language. The more recent *Uby* effort aims at aligning concepts from different sources such as WordNet, GermaNet, FrameNet, Wiktionary and Wikipedia. It comprises more than 4.2 million lexical entries and 0.75 million links that were both manually and automatically generated (using mapping algorithms) [10]. Both BabelNet and Uby are useful resources, although they still restrict themselves on concepts and entities already listed in the existing sources. We aim at a more general approach for extracting semantic concept relations from unstructured text, even for concepts that are not yet listed in an existing repository such as WordNet.

In 1992, Marti A. Hearst proposed the use of lexico-syntactic patterns to extract synonym and hyponym relations in unrestricted text, like "A is a form of B" (A is-a B) or "A_1, ..., A_{n-1} and other A_n" ($A_1, ..., A_n$ are synonyms) [14]. In [15], such *Hearst patterns* are used to create ontologies from

[1] http://wordnet.princeton.edu/

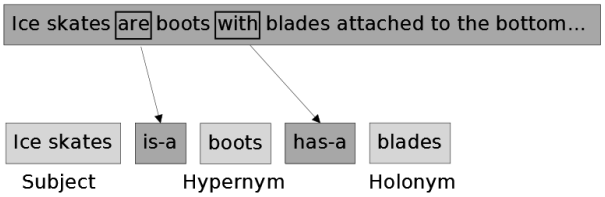

Figure 1: Sample sentence containing two semantic relation patterns.

Wikipedia pages. The approach focuses on the biological domain and can handle only simple semantic patterns. They obtain a rather poor recall (20 %) but excellent precision (88.5 %).

In [26], [25], Ruiz-Casado and colleagues apply machine learning to learn specific Hearst patterns in order to extract semantic relations from Simple Wikipedia and link them to WordNet. As they only link words (nouns) to WordNet concepts, they are facing the same coverage problem as mentioned for Yago. Simple Wikipedia has a quite restricted content, leading to only 1,965 relationships, 681 of which are already part of WordNet. Snow et al. [27] also apply machine learning to learn Hearst patterns from news texts in order to decide whether words are related by hypernyms or hyponyms. In [7], the authors introduce a supervised learning approach to build semantic contraints for part-of relations in natural text. Those patterns are retrieved by using a selection of WordNet part-of relations as training data, which are gradually generalized and disambiguated.

Sumida and Torisawa focus on finding hyponymy relations between concepts from the Japanese Wikipedia [30]. They exploit the internal structure of Wikipedia pages (headings, sub-headings, sub-sub-headings etc.) together with pattern matching and different linguistic features. They could retrieve 1.4 million relations with a precision of about 75 %. Ponzetto and Strube [20] also exploit the category system and links of Wikipedia to derive is-a and non is-a relations by applying lexico-syntactic pattern matching.

In our approach, we will also apply semantic patterns to determine semantic relations similar than in the previous approaches. However, we focus more on the actual text of Wikipedia articles (especially Wikipedia definitions) rather than on the existing category system, info boxes or hyperlinks between pages. Also, we are especially interested in conceptual relations (as opposed to links between named entities) and try to cover not only hyponym (is-a) relations, but also equal, part-of and has-a relations.

3. SEMANTIC RELATION PATTERNS

Semantic relation patterns are the core features in our approach to find semantic relations. We focus on their identification in the first sentence of a Wikipedia article which mostly defines a concept or term and thus contains semantic relations. The sample sentence in Fig. 1 contains two semantic patterns defining 'ice skates'. In this section, we introduce the notion of semantic patterns and discuss different variations needed in our approach. In the next section, we describe in detail the use of semantic patterns for finding semantic relations.

Table 2: Typical patterns for is-a relations (hyponyms).

Hypernym patterns
is a
is typically a
is any form of
is a class of
is commonly any variety of
describes a
is defined as a
is used for any type of

Table 3: Typical patterns for part-of relations (meronyms) and has-a relations (holonyms).

Meronym patterns	Holonym patterns
within	consists/consisting of
as part of	having
in	with
of	

A semantic relation pattern is a specific word pattern that expresses a linguistic relation of a certain type (like hyponym resp. is-a). It connects two sets of words X and Y appearing left and right of the pattern, much like operands of a comparison relationship. There are general patterns for hyponym (is-a) relations, meronym (part-of) relations, holonym (has-a) relations and synonym (equal) relations, the is-a patterns being the most commonly occurring ones in Wikipedia definitions. For example, the simple pattern "is a" in *"A car is a wheeled motor vehicle."* links the concepts *car* and *vehicle* by a hyponym relation. Having these two concepts and the semantic relation pattern, we can build the semantic relation (car, is-a, vehicle). The example in Fig. 1 shows that there may be more than one semantic pattern in a sentence that need to be correctly discovered by our approach.

3.1 Is-a patterns

According to our experiences, "is-a" patterns occur in versatile variations and can become as complex as "X is any of a variety of Y". They appear often with an additional (time) adverb like *commonly*, *generally* or *typically* and expressions like *class of*, *form of* or *piece of* (collectives and partitives). They can appear in plural and singular ("is a" or "are a") and come with different determiners (like is a/an/the) or no determiner at all as in the *ice skates* example. They invariably come with a verb, but are not necessarily restricted to the verb *be*. Table 2 shows some examples of frequently occurring is-a patterns that we use in our approach. The list of patterns is extensible so that a high flexibility is supported.

3.2 Part-of / Has-a Patterns

Typical patterns for part-of and has-a relations are shown in Table 3. The adverb *within* and the prepositions "in" and "of" often indicate part-of relations, e.g., for *"A CPU is the hardware within a computer."*, leading to (CPU, part-of, computer), and for *"Desktop refers to the surface of a desk"*, leading to the correct relation (desktop, part-of, desk). How-

Table 4: Typical synonym patterns in itemizations.

Synonyms patterns
A, B and C
A, also called B
A, also known as B or C
A, sometimes also referred to as B

ever, these patterns can also be misleading, as such prepositions can be used in various situations, as *"Leipzig University was founded in the late Middle Ages."*, which would lead to the not really useful relation (Leipzig University, part-of, Middle Ages). Similar arguments hold for holonym patterns, where *consisting of* is often more reliable than the rather diversely used words *having* and *with*. Valid examples include *"A computer consists of at least one processing element"*, leading to (processing element, part-of, computer) and the ice skates example resulting in (blades, part-of, ice skates). On the other hand, *"A screw-propelled vehicle is a land or amphibious vehicle designed to cope with difficult snow and ice or mud and swamp."* is a misleading case, as it can lead to relations like *"snow, part-of, screw-propelled vehicle"*.

3.3 Equal Patterns

Finally, Table 4 shows some constructions for synonym relations. In itemizations occurring before another semantic pattern, the terms they comprise are generally synonyms (as in *"A bus (archaically also omnibus, multibus, or autobus) is a road vehicle"*). Outside itemizations, there are also a few binary synonym patterns like "is a synonym for", "stands for" (in acronyms and abbreviations) or "is short for" (in shortenings). They are quite rare in Wikipedia, as synonym words are typically comprised in exactly one page (for example, there is only one Wikipedia page for the synonym terms *car*, *motor car*, *autocar* and *automobile*). Thus, instead of a definition like *"A car is a synonym for automobile"* articles rather look like *"An automobile, autocar, motor car or car is a wheeled motor vehicle [...]"*. In this case, four synonym terms are related to one hypernym term (*wheeled motor vehicle*). Our approach is able to identify multiple semantic relations in such cases.

4. DISCOVERING SEMANTIC CONCEPT RELATIONS

This section outlines in detail how we extract semantic concept relations from Wikipedia. The overall workflow is shown in Fig. 2. We start with a preparatory step to extract all relevant articles from Wikipedia. For each article we perform the following four sub-steps:

1. For each article, we perform some preprocessing to extract its first sentence (the "definition sentence") and to tag and simplify this sentence.

2. In the definition sentence, we identify all semantic relation patterns. If there are n such patterns ($n \geq 1$), we split the sentence at those patterns and thus obtain $(n + 1)$ sentence fragments. If there is no pattern, we skip the article.

3. In each sentence fragment, we search for the relevant

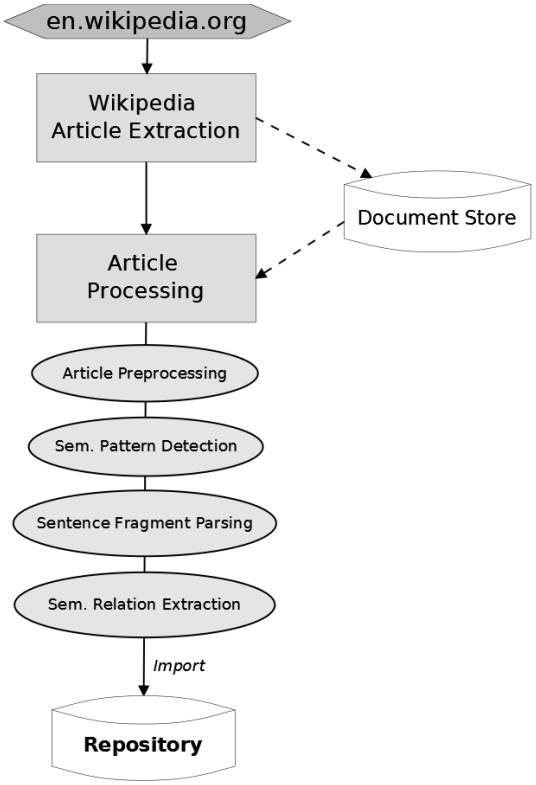

Figure 2: Workflow to extract semantic relations from Wikipedia.

concepts that are linked by the semantic relation patterns.

4. Having the terms and patterns, we build the respective semantic relations and import them in our repository.

The workflow is carried out automatically, i.e., no human interaction is required. It uses a few manually created resources, like a list of typical English partitives (e.g., *kind of*, *type of*, *genus of*) and anchor terms for the pattern detection, but apart from that it does not need any additional background sources.

For our example sentence of Fig. 1, we would after preprocessing identify in the second step the two semantic patterns "is-a" and "has-a" and determine three sentence fragments. We also find out that *ice skates* is the subject (left operand) for both semantic relations while the other fragments refer to the second operand (object) for the relations. The fragments are further processed in the third step where we determine that the hypernym concept is *boots* and the holonym concept is *blades*. We finally derive the two semantic relations (ice skates, is-a, boots) and (ice skates, has-a, blades) and store them in the repository.

In the following, we first describe the preparatory extraction of articles from Wikipedia and then outline the four major steps to be performed per article.

4.1 Extracting Wikipedia Articles

The easiest way to access the full Wikipedia content is to download the Wikipedia dump, which is basically a single XML file containing all articles with the respective content and meta information (like page id, creation date etc.).[2] This file contains about 11 million entries and has an overall size of 44 GB (unzipped). However, the English Wikipedia comprises only 4.45 million articles (as of February 2013), as there are many additional pages listed in the Wikipedia dump which are no articles, like category pages, redirects, talk pages and file pages (images, charts etc.).

The first step of our approach is to extract each Wikipedia article name together with the abstract section of the article. We will carefully remove the aforementioned pages that do not represent Wikipedia articles. As it is partly difficult to determine the abstract of an article (which is the section occurring before the table of contents), and as some articles simply do not contain abstract and main section, we extract the first 750 characters in each article. The ratio behind this limit of 750 characters is that we are currently only parsing the first sentence of each Wikipedia article, which is typically a definition sentence containing the relevant information. We may try parsing the second or third sentence later on, because they occasionally contain some additional information, but currently do not intend to parse the full text of Wikipedia articles, so that the first 750 characters of each article will suffice for our purposes.

The extracted text is subsequently cleaned from Wikipedia-specific formatting commands using the Java Wikipedia API (Bliki engine)[3], and then page name and extracted text are stored as documents in a document database (MongoDB) for further processing.

[2]http://en.wikipedia.org/wiki/Wikipedia:Database_download

[3]http://code.google.com/p/gwtwiki/

Table 5: POS-tagging example for the *Ice skates* article.

Word	POS tag	Word class
Ice	NN	noun (singular)
skates	NNS	noun (plural)
are	VBP	verb (plural)
boots	NNS	noun (plural)
with	IN	preposition
blades	NNS	noun (plural)
attached	VBN	verb (past participle)
to	TO	"to"
it	PRP	personal pronoun

4.2 Article Preprocessing

Before we start parsing the definition sentence of a Wikipedia article, we perform some textual preprocessing. We first replace intricate expressions that cannot be handled by our approach (*sentence simplification*). Such expressions will be replaced by simpler expressions or, in specific cases, removed entirely. For instance, we replace the rare and bulky expression "is any of a variety of" by "is any" which can be handled well by our approach. The advantage of such simplifications is that it avoids a more complex processing in later steps without losing information.

Secondly, we perform Part-of-Speech tagging (POS tagging) using the Apache OpenNLP Library for Java[4]. The POS tagger determines the word class of each word in the sentence and annotates the words accordingly. After this, the sentence "Ice skates are boots with blades attached to it." looks as follows:

Ice_NN skates_NNS are_VBP boots_NNS with_IN blades_NNS attached_VBN to_TO it_PRP.

Table 5 gives a clearer representation of the sentence together with the POS tags and their meaning.

4.3 Identifying Semantic Relation Patterns

To identify semantic relation patterns, we parse the first sentence of an article word by word and apply a finite state machine (FSM) to discover these patterns. Fig. 3 shows a simplified version of the FSM for the is-a patterns, consisting of nine states. The dark gray states $(1, 2)$ represent initial states for different terms, so-called *anchor terms*, indicating the beginning of an is-a pattern. Anchor terms can be in singular (like *is*, *describes*) or plural (like *are*, *describe*). If we find any anchor term, we continue processing the FSM. Starting from either of the two initial states, we check the following word or sequence of words in the sentence and can thus change into another state (transition). For instance, if the word after an anchor term is an adverb like "commonly", we change into state 3. If we reach the final state (9), we have detected the entire pattern. We will then go on with the word-by-word parsing to look for another semantic relation pattern. For any newly found anchor term, we process the corresponding FSM until we have found all semantic patterns.

Some transitions have additional conditions, like $I \neq 1$, meaning that this transition can only be used if the initial

[4]http://opennlp.apache.org/

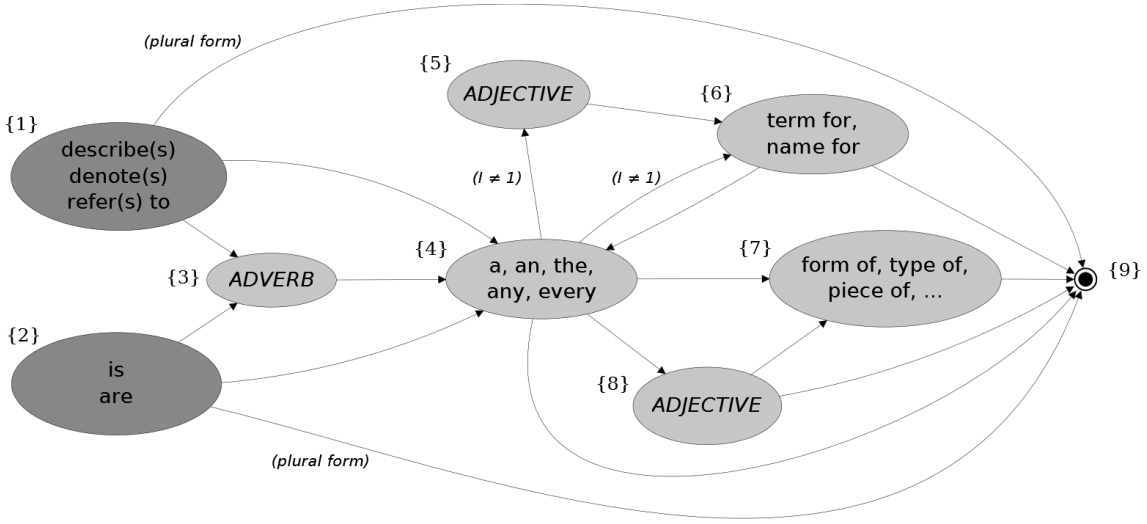

Figure 3: FSM for parsing is-a patterns (simplified).

state was not 1, or "plural form", meaning that this transition can only be used if the anchor term has the plural form. For example, "Ice skates **are** boots" uses this case, in which the state path is simply $(2, 9)$. If the verb is in singular, we normally have at least one determiner (covered by state 4). The states passed in the FSM can become more complex, as in the example *"A baker's rack* ***is a type of*** *furniture"*, which has the state path $(2, 4, 7, 9)$ or *"Adenitis* ***is a general term for an*** *inflammation of a gland."* $(2, 4, 5, 6, 4, 9)$.

Conditions for transitions into another state can become rather complex; they are mostly excluded from Fig. 3 for better legibility. In many cases we have to check the part of speech (like adjective, adverb, determiner) of the next word, or we have to check the next word as such. In state 4, for instance, there are two adjective states we could enter (5 or 8). Thus, to change into state 5, we have to check that the next word is an adjective, and the further two words are "term for" or "name for". We also have to check that the initial state has not been 1, as phrases like "A describes a term for B" are insensible, while "A is a term for B" is correct (initial state 2). If no transition is possible, we cannot reach the final state and leave the FSM. We then continue parsing the sentence to possibly find another anchor term.

The finite state machines we use were manually developed. We started with a basic implementation to handle simple patterns like A is a B, and then iteratively fine-tuned our approach to cover more specific cases. To determine these specific cases we processed several hundreds of randomly selected articles, searching for Wikipedia definitions that could not be processed. The revised FSMs are now able to handle most of the articles in Wikipedia.

Most article definitions contain exactly one pattern, namely a hyponym (is-a) pattern. Sometimes, an additional meronym or holonym pattern is also available, generally succeeding the is-a pattern. Less frequently, no hyponym pattern is available, but a meronym or holonym pattern. We thus obtain mainly the following three combinations, where A, B and C are lists of terms:

1. $A \xrightarrow{Hyponym} B$
2. $A \xrightarrow{Hyponym} B \overset{Hol./Mer.}{\hookrightarrow} C$
3. $A \xrightarrow{Hol./Mer.} B$

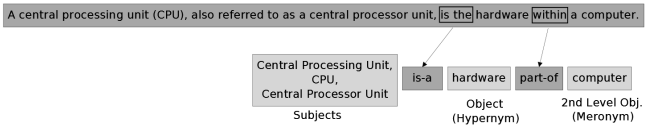

Figure 4: Sample definition sentence with two patterns, 3 subject terms, 1 object term (hypernym) and 1 second-level object (meronym).

The hooked arrow in case 2 denotes that this pattern links A and C, but not B and C (as one might expect at first glance). Loosely related to the linguistic field of syntax, we call A the *subjects* and B the *objects* in the definition sentence (the patterns would be the verb then). If there is a C, we call C the *second-level objects* for disambiguation. Fig. 4 gives an example of a Wikipedia sentence with two patterns, three subjects, one object and one second-level object. We split the sentence at the patterns and extract the subject, object and secondary object fragment for further processing in the next step.

If we find two relation patterns P and P' in a sentence, we use an individual threshold L (default value 7) specifying that at most L words may be between P and P'. If the number of words in between exceeds L, we reject P' and only use P. This strict regulation became necessary since we observed in test cases that patterns occurring in such a distance from the first pattern are frequently incorrect, e.g., meronym and holonym patterns for simple prepositions like "in" and "of".

4.4 Parsing Sentence Fragments

The sentence fragments representing subjects, objects or secondary objects need to be processed to identify the con-

cept (term) or list of concepts that participate in a semantic relation. For this purpose, we apply a further FSM. In many cases, the nouns directly left and right from a semantic relation pattern represent already relevant concepts, thus allowing for a simple extraction. However, the following examples illustrate that such a strategy is generally too simple to correctly extract the relevant concepts:

1. *"A wardrobe, also known as an armoire from the **French**, is a standing **closet**."* (French is a closet)
2. *"Column or pillar in **architecture** and **structural engineering** is a **structural element**."* (architecture and structural engineering are structural elements)

The first example contains some additional etymological information, which can impair the extraction of subject terms. The second example contains a *field reference* that describes in which (scientific) field or domain the article is used, or if it is a homonym, to which field or domain the current page refers to. There is no general relation between field terms and semantic relation patterns, but between field terms and subject terms. Thus, a subject term is normally "found" in the specified field or domain, which suggests the part-of relation. In example 2, we would thus conclude "column and pillar are part of architecture and structural engineering). Therefore, we extract both field references and subject terms.

It is especially important to retrieve all relevant concepts, which is difficult as some appear in additional parentheses (where also irrelevant terms may be found) or appositions. The FSM to process a single sentence fragment is therefore rather voluminous. The subject fragment parsing FSM alone contains about 20 states (5 initial states) and more than 40 transitions. There are subordinate FSMs that take control over specific tasks, such as to decide in an itemization of concepts where to start and where to stop, and whether an adjective or verb is part of a compound concept (like high school) or whether it is a word that does not belong to the actual concept (like a *wheeled vehicle*, in which *wheeled* is not part of the actual concept we would like to extract), although this is not always unequivocally possible to decide.

Parentheses are quite difficult to handle. It would be rather easy to simply remove all parentheses expressions, as they often contain only augmenting information, but can lead to insensible extractions or, more likely, bring the FSM into an illegal state. On the other hand, parentheses often contain synonyms which are very important for our purposes. We thus decided to run our approach in different configurations for every article. We first try to parse it without touching the parenthesis expression. If the article cannot be successfully parsed, we replace the parenthesis by commas and turn them into a real apposition. We also have a similar configuration in which the left parenthesis is replaced by ", or" and the right parenthesis by a comma. For instance, *"An auto (automobile) is a..."* would be converted into *"An auto, or automobile, is a..."*, which is an expression the FSM can easily handle. Finally, if the parsing still fails, we remove the entire parenthesis expression. We risk to miss some synonyms then, but may be able to successfully parse the sentence after all.

4.5 Determining Semantic Relations

Once the subject terms, object terms and field terms have been extracted, we build the semantic relationships. The outcome of each successfully parsed article is a set of (1:1)-relationships in which two terms are related by one semantic relation. There is one important aspect: The subjects are all related to the objects by the semantic relation patterns, but as they are synonyms, they are also related to each other by an equal relation. Hence, the sentence *"An automobile, autocar, motor car or car is a wheeled motor vehicle [...]"* results into four is-a relations as well as six equal relations for the four synonyms.

The equal relation does generally not hold between different objects as the following example shows: *"A rocket is a missile, spacecraft, aircraft or other vehicle that obtains thrust from a rocket engine."* Although the four objects *missile*, *spacecraft*, *aircraft* and *vehicle* are all related to each other, they are not truly equivalent. This is a typical situation so that we do not derive equal relations between different object terms.

Let $|S|$ be the number of subjects, $|O_1|$ the number of objects and $|O_2|$ the number of second-level objects. The number of synonym relations R_s is:

$$|R_s| = \binom{|S|}{2} = \frac{|S| * (|S| - 1)}{2} \quad (1)$$

The number of relations between S and O is $S * O$, since any subject is related to any object. The same holds for the number of relations between S and F (the field references). We thus have the following number of one-to-one relations $|R|$:

$$|R| = \binom{|S|}{2} + (|S| * |O_1|) + (|S| * |O_2|) + (|S| * |F|) \quad (2)$$

Note that this number can become rather large. For instance, for 5 subject terms, 2 hypernym objects, 2 meronym objects and 1 field reference, we obtain 35 semantic relations from just one Wikipedia article. Although this is a rare case, it illustrates the richness of our strategy compared to previous approaches that only link Wikipedia page names with other resources like WordNet, leading to at most one link per Wikipedia article.

All determined semantic relations are finally added to a repository. They can then be used as background knowledge, e.g. for semantic ontology matching. Our approach also comprises a filter that determines whether a Wikipedia article refers to a named entity or a concept. This filter will primarily check whether the extracted source concepts of an article start with a capital letter, which is a strong hint for a named entity like a city, region, person or company. We will apply this filter when we integrate the extracted relations in the repository to restrict the background knowledge on conceptual relations.

5. EVALUATION

In our evaluation we analyze the effectiveness of the proposed approach for four different subsets of Wikipedia covering different subjects and domains. We evaluate the different substeps to determine the semantic relation patterns and to determine the subjects and objects as well as the overall effectiveness regarding semantic relations. In the following, we first describe the used benchmark datasets. We then focus on the effectiveness of the FSMs to parse articles for find-

Table 6: Benchmark datasets with their number of articles W and number of parsable articles W_p.

Name	Domain	W	W_p
Furniture (F)	General	186	169
Infectious Diseases (D)	Medicine	107	91
Optimization Algorithms (O)	Mathematics	122	113
Vehicles (V)	General	94	91

ing semantic relation patterns. In subsection 5.3 we analyze the effectiveness of concept extraction to identify subjects, objects and secondary objects. We then evaluate how many concepts (subjects, objects and field references) could be retrieved from the Wikipedia articles and how many of them were correct. Subsequently, we evaluate how many relations we could generate from the patterns and concepts extracted, and how many of them were correct. We close this chapter with some observations on problem cases that could not yet be handled sufficiently.

5.1 Benchmark Datasets

To evaluate our approach we have chosen four sets of Wikipedia articles as benchmark datasets. Each such article set consists of all articles in a specific category, with the exception of "List of" articles that we neglected (as they never contain any semantic relations). We tried to use categories from different domains and with a representative number of articles W (we aimed at benchmarks containing around 100 articles). Categories often contain sub-categories, which we did not include, though.

Wikipedia categories are rather heterogeneous, which makes the benchmark datasets quite interesting. For instance, in the furniture category there are both very general concepts (like *couch*, *desk*, *seat*) and specific concepts (like *cassone*, *easel*, *folding seat*). By contrast, some general concepts one would definitely expect in the furniture category are not listed there, such as *chair*, *table* or *bed* (although *Windsor chair*, *sewing table* and *daybed* are listed). Typically, those concepts have a separate sub-category then.

Table 6 gives an overview of the datasets we use in our evaluation. The datasets *furniture*[5], *infectious diseases*[6] and *optimization algorithms*[7] refer to category pages while *vehicles*[8] is based on an outline page (which is similar to a list, but represented as a Wikipedia article). The datasets were generated in October 2013 and their articles may slightly differ from current versions due to some recent changes.

It turned out that not all Wikipedia articles in the datasets could actually be used for our purposes, since they include articles that do not have a classic definition using a hypernym, holonym or meronym. These articles do not contain any specific semantic relation pattern and are thus not parsable. Examples of such non-parsable articles include:

- *Anaerobic infections are caused by anaerobic bacteria.*
- *Hutchinson's triad is named after Sir Jonathan Hutchinson (1828 - 1913).*
- *A diving chamber has two main functions: as a simpler form of submersible vessel to take divers underwater and to provide a temporary base and retrieval system [...]*

We exclude such articles from our evaluation and only consider the parsable articles W_p. The number of these articles is shown in the last column of Table 6.

[5]http://en.wikipedia.org/wiki/Category:Furniture
[6]http://en.wikipedia.org/wiki/Category:Infectious_diseases
[7]http://en.wikipedia.org/wiki/Category:Optimization_algorithms_and_methods
[8]http://en.wikipedia.org/wiki/Outline_of_vehicles

Table 7: Evaluation of pattern detection.

	W_p	ω	ω^T	r_{pars}	r_{rel}	p_{rel}
F	169	148	142	.88	.84	.96
D	91	80	80	.88	.88	1
O	113	84	84	.74	.74	1
V	91	87	87	.96	.96	1

5.2 Article Parsing and Pattern Discovery

In this section, we evaluate how many articles we were able to fully parse using our FSMs. This includes the detection of at least one semantic relation pattern in the definition sentence, the sentence fragmentation and the extraction of at least one subject and one object. We use the classic recall and precision measures to evaluate our approach:

Given a set of parsable Wikipedia articles W_p, let ω be the number of articles we could successfully parse, and ω^T the number of articles where the correct semantic relation pattern was detected. We determine the recall (accuracy) for parsing r_{pars}, the recall for finding the correct semantic relation pattern r_{rel}, and the precision for finding semantic relation pattern p_{rel} as follows

$$r_{pars} = \frac{\omega}{W_p} \qquad r_{rel} = \frac{\omega^T}{W_p} \qquad p_{rel} = \frac{\omega^T}{\omega} \tag{3}$$

Table 7 shows for each benchmark dataset the number of parsable articles (W_p), the number of parsed articles (ω), the number of correctly detected patterns (ω^T) as well as parsing recall r_{pars}, relation recall r_{rel} and relation precision p_{rel}. Our approach can parse between 74 % and 96 % (86 % on average) of all Wikipedia articles that contain any semantic relations. Parsing the scientific articles in dataset O tends to be more error-prone (74 %) than rather general articles (F, V) with 88-96% parsing recall.

The semantic patterns were in most cases correctly detected, leading to a precision of 96-100 %. With the exception of only one article, the first pattern discovered in the definition sentence was invariably a hyponym pattern. If there was a second semantic pattern, it was invariably a holonym or meronym pattern.

5.3 Term Extraction

Now that we have demonstrated how many articles could be parsed and in how many cases we detected the correct semantic pattern, we evaluate how many of the terms (subjects, objects and fields) encoded in the articles of the benchmark datasets were discovered (recall), and how many of the extracted terms were correct (precision). These terms, together with the semantic patterns (relations) make up the overall semantic relations we can derive from the Wikipedia article sets (we discuss these relations in the next subsection).

Table 8: Recall and precision for term extraction.

	r				p			
	S.	O.	2L O.	F.	S.	O.	2L O.	F.
F	.93	.90	.66	.8	.95	.87	.58	.73
D	.85	.87	.70	1	.96	.80	.57	.67
O	.84	.91	.81	1	.88	.88	.36	.92
V	.83	.94	.86		.96	.94	.49	

Table 9: Number of extracted concepts and relations from each benchmark.

	W_P	Subj.	Obj.	2L O.	Fields	Rel.
F	169	200	142	43	4	373
D	91	111	58	26	4	206
O	113	84	66	6	23	137
V	91	138	78	17	0	280

We denote T^P as the terms that occur in Wikipedia pages which were parsed by our approach. We denote T_C as the correctly identified terms and T_F as the falsely identified terms. Again, we use recall and precision to assess our approach:

$$r = \frac{T_C}{T^P} \tag{4}$$

$$p = \frac{T_C}{T_C + T_F} \tag{5}$$

Table 8 shows recall and precision of the term extraction. We provide results for all types of terms we extracted, i.e., subjects (S), objects (O), second-level objects (2L O) and field references (F). The recall is similarly good for all datasets and about 83 to 94 % of the subjects and first-level objects could be correctly extracted. Extracting the second-level objects (has-a or part-of relations) is more difficult and ranges from 66 to 86 %.

Precision is a little higher for subjects, where 88 to 96 % of the extracted concepts where correct, while only 80 to 94 % of the extracted first-level objects were correct. Extracting the second-level objects is more error-prone. We achieve only 36 to 58 % precision, meaning that a considerable amount of terms are extracted which are actually no concept being part in any relation.

Field references occurred only scarcely in the considered benchmark datasets. There was no field reference in V, which is quite natural, as there is no "field of vehicles". In the other scenarios, we found most of the field references (80-100 %) with a precision ranging between 67 and 92 %.

Table 9 shows the number of articles in each benchmark and the number of terms we could correctly extract. The number of subjects is always highest, because many synonyms are found in the article definitions. The number of first-level objects is lower, as we find generally only one object (hypernym) in the definition. The number of second-level objects is again lower, as meronym and holonym relations occur only occasionally. The last column describes the number of correct relations we could derive from each benchmark. Comparing this number with the number of articles in the benchmark, it can be seen that we are able to extract an average amount of 1.2 to 3.1 relations per article (including articles we failed to process).

Table 10: Number of relations per benchmark, correctly extracted relations, falsely extracted relations as well as recall and precision.

	Rel. in W_p	Correct rel.	False rel.	r	p
F	497	373	87	.75	.81
D	323	206	67	.64	.76
O	182	137	49	.76	.74
V	413	280	66	.68	.81
$\sum$	1,415	996	269	.70	.79

Table 11: Distribution, recall and precision for each individual relation type in the Furniture benchmark.

	share	r	p
equal	24.1 %	.73	.87
is-a	55.4 %	.87	.88
has-a / part-of	19.4 %	.58	.63
field ref.	1.1 %	.36	.57

5.4 Evaluation of Semantic Relations

We have demonstrated how many semantic patterns we could correctly detect, and how many subject terms, object terms and field references we could extract. As a last step, we have to put these terms and patterns together to obtain a set of semantic relations. We will now show how many of these relations we could derive from each benchmark dataset, and how many of them were correct.

Table 10 presents these final results of our approach. It contains the number of overall relations contained per benchmark, as well as the number of correctly and falsely extracted relations with the corresponding recall and precision values. We can extract 64 to 76 % of all relations that are in the benchmark with an average recall of 70 %. Precision is slightly higher, ranging between 74 and 81 % with an average of 79 %.

Finally, we present the detailed results for each relation type in Tab. 11, which we performed on the Furniture benchmark. The first column specifies how often each individual relation type occurred (we computed the value on the correctly extracted relations). As it can be seen, more than half of all extracted relations are is-a relations. Equal-relations (24 %) and has-a resp. part-of relations (19 %) occur less often while field references are extremely rare (1 %). Regarding the quality of each type, is-a relations have the best recall and precision, while equal-relations have a similar precision but somewhat lower recall; has-a resp. part-of relations only achieve moderate results both in recall and precision. Similar observations can be made w.r.t. the field references, although this value is difficult to judge, as there were only 4 field references in the benchmark.

5.5 Observations

The evaluation results show that the proposed approach is already highly effective as it correctly parsed 74 % to 98

% of all parsable articles and retrieved approx. 70 % of all relations in the 4 benchmarks with an approx. precision of 79 %. Still, we were not able to detect all relevant relations, nor could we prevent erroneous extractions. To explain and illustrate the reasons for extraction errors we discuss some of the observed problem cases in the following.

Parsing errors are often caused by complex sentence structures, especially for complex introductions or subjects. Examples of articles that could not be successfully parsed include:

1. *Cryptic Infections: an infection caused by an as yet unidentified pathogen...*
2. *A hospital-acquired infection, also known as a HAI or in medical literature as a nosocomial infection, is an infection...*
3. *Lower respiratory tract infection, while often used as a synonym for pneumonia, can also be applied to other types of infection including...*

Example 1 does not contain any semantic relation pattern and uses the dictionary notation that is also used in Wiktionary but uncommon in Wikipedia articles. The two other examples have too complex subject fragments leading the parser into an illegal FSM state. A more advanced FSM might be able to handle these cases, but being very specific Wikipedia definitions, would have only little impact on the overall recall.

Regarding term extraction, recall problems are typically caused by complex expressions with parentheses that have to be removed in order to successfully parse the sentence. If such parentheses contain relevant terms, they will not be extracted.

The POS-tagging is also error-prone, as in the following snippet *"A dog sled is a sled used for..."*, where both occurrences of *sled* are tagged as verbs, although the noun is meant. Our FSM does not expect a verb before or after the is-a pattern, so would refuse the article. In an extended version of our system, we also consider the page name of the article to be parsed. If a word in the sentence appears in the page name (as *sled* in the example is part of the article name *dog sled*), we accept the word even if it has an unreasonable word class. Still, this cannot avoid all abortions caused by erroneous POS-tagging.

Precision of term extraction is, among others, impaired by the following reasons:

- The parser draws much on the POS tags of the words in a sentence. In complex words, it may be misled, as in the following example: *A minibus is a passenger carrying motor vehicle.* The extracted relation is "A minibus is a passenger", as it does not discover the actual compound word "passenger carrying motor vehicle".
- Similarly, compounds cannot always be correctly determined. For instance, in *"A draisine is a light auxiliary rail vehicle"* the object *"light auxiliary rail vehicle"* is extracted. However, the actual compound would be *rail vehicle*, while *light auxiliary* is an attribute. The parser cannot generally ignore adjectives (and participles), as some compound words contain them (like *high school*, *bathing suite*, *pulled rickshaw*).
- Sometimes, itemizations contain misleading nouns as in *"Jet pack, rocket belt, rocket pack and similar names are used for various types of devices"*, where "similar names are devices" is extracted. Similar arguments hold for expressions like *"is a noun for"*, *"is the act of"* etc., which are all relatively rare. We will extend our list of simplifications to avoid such pitfalls.

We also compared our extracted results with the information encoded in the Wikipedia category system. Although many relations are found in the category system (like *table* is listed under the category of *furniture*), we were also able to extract is-a relations not provided by the category system. For instance, we extracted that *paint* is a *liquid*, but the paint article is only listed under the concept of paints, and we could not find any link between liquids and paints in the category system. Furthermore, the category system only provides is-a relations, while we can also extract has-a and part-of relations. Eventually, we can extract the various synonyms for a term described by an article, which can only partly be retrieved using the re-direct pages in Wikipedia, as such pages may not necessarily cover all synonyms found in the article definition.

6. OUTLOOK AND FUTURE WORK

We proposed and evaluated a novel approach to extract semantic concept relations from unstructured Wikipedia articles. The approach focuses on the analysis of the definition sentence of Wikipedia articles and uses finite state machines to extract semantic relation patterns and their operands to discover semantic relations. The approach is flexible and can find several semantic relations of different types (is-a, part-of, has-a, equal) per article. The evaluation showed the high effectiveness of the approach for different domains.

In future work, we will use the approach to build up a comprehensive repository of semantic concept relations and use this repository for semantic ontology matching. We further plan to extract semantic relations from additional sources such as "Wiktionary" and combine the new repositories with other repositories such as WordNet. Furthermore, we can post-process derived relations to verify their correctness and derive additional relations, e.g., by using linguistic techniques as we already presented in [1]. For instance, with a strategy for compounds we could derive from the relation (draisine, is-a, light auxiliary rail vehicle) the related relations (draisine, is-a, auxilary rail vehicle) and (draisine, is-a, rail vehicle).

7. ACKNOWLEDGMENTS

This study was partly funded by the European Commission through Project "LinkedDesign" (No. 284613 FoF-ICT-2011.7.4).

8. REFERENCES

[1] Patrick Arnold and Erhard Rahm. Semantic enrichment of ontology mappings: A linguistic-based approach. In *Advances in Databases and Information Systems*, LNCS 8133, pages 42–55. Springer, 2013.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *6th Int. Semantic Web Conference(ISWC)*, LNCS 4825, pages 722–735. Springer, 2007.

[3] Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors. *Schema Matching and Mapping*. Springer, 2011.

[4] J. Euzenat and P. Shvaiko. *Ontology Matching (2nd ed.)*. Springer, 2013.

[5] Tiziano Flati and Roberto Navigli. Spred: Large-scale harvesting of semantic predicates. In *Proc. 51st Annual Meeting of the Association for Computational Linguistics*, pages 1222–1232, 2013.

[6] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. 20th Int. Joint Conf. on Artificial Intelligence*, pages 1606–1611, 2007.

[7] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proc. Conf. North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL '03, pages 1–8. Association for Computational Linguistics, 2003.

[8] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-Match: An Algorithm and an Implementation of Semantic Matching. In *Proc. 1st European Semantic Web Symposium*, LNCS 3053, pages 61–75. Springer, 2004.

[9] Anika Groß, Júlio Cesar dos Reis, Michael Hartung, Cédric Pruski, and Erhard Rahm. Semi-automatic adaptation of mappings between life science ontologies. In *Proc. 9th Int. Conf. on Data Integration in the Life Sciences (DILS)*, Lecture Notes in Bioinformatics (LNBI) 7970, pages 90–104. Springer, 2013.

[10] Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth. Uby: A large-scale unified lexical-semantic resource based on LMF. In *Proc. 13th Conf. European Chapter of the Association for Computational Linguistics*, EACL '12, pages 580–590. Association for Computational Linguistics, 2012.

[11] Willem Robert Van Hage, Sophia Katrenko, and Guus Schreiber. A method to combine linguistic ontology-mapping techniques. In *Proc. Int. Semantic Web Conference (ISWC)*, LNCS 3729, pages 732–744. Springer, 2005.

[12] Faycal Hamdi, Brigitte Safar, Nobal B. Niraula, and Chantal Reynaud. Taxomap alignment and refinement modules: results for OAEI 2010. In *Proc. 5th Intern. Workshop on Ontology Matching (OM)*, CEUR Workshop Proceedings 689, 2010.

[13] Michael Hartung, James F. Terwilliger, and Erhard Rahm. Recent advances in schema and ontology evolution. In *Schema Matching and Mapping*, pages 149–190. Springer, 2011.

[14] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc . 14th Conf. on Computational Linguistics*, COLING '92, pages 539–545. Association for Computational Linguistics, 1992.

[15] Aurelie Herbelot and Ann Copestake. Acquiring ontological relationships from Wikipedia using RMRS. In *Proc. ISWC 2006 Workshop on Web Content Mining with Human Language Technologies*, 2006.

[16] Sabine Maßmann, Salvatore Raunich, David Aumüller, Patrick Arnold, and Erhard Rahm. Evolution of the COMA match system. In *Proc. 6th Intern. Workshop on Ontology Matching (OM)*, CEUR Workshop Proceedings 814, 2011.

[17] George A. Miller. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41, November 1995.

[18] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 216–225, 2010.

[19] Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1522–1531, 2010.

[20] Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from Wikipedia. In *Proc. 22nd National Conf. on Artificial Intelligence*, AAAI'07, pages 1440–1445, 2007.

[21] E. Rahm. Towards Large Scale Schema and Ontology Matching. In *Schema Matching and Mapping*, chapter 1, pages 3–27. Springer, 2011.

[22] E. Rahm and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 10:334–350, 2001.

[23] Salvatore Raunich and Erhard Rahm. ATOM: Automatic target-driven ontology merging. In *Proc. 2011 IEEE 27th Int. Conf. Data Engineering*, ICDE '11, pages 1276–1279, 2011.

[24] Salvatore Raunich and Erhard Rahm. Target-driven merging of taxonomies with Atom. *Information Systems*, 42(0):1 – 14, 2014.

[25] Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *Proc. 10th Int. Conf. Natural Language Processing and Information Systems*, NLDB'05, pages 67–79. Springer, 2005.

[26] Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data & Knowledge Engineering*, 61(3):484 – 499, 2007.

[27] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[28] Michael Strube and Simone Paolo Ponzetto. Wikirelate! Computing semantic relatedness using Wikipedia. In *Proc. 21st National Conf. on Artificial Intelligence*, pages 1419–1424. AAAI Press, 2006.

[29] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proc. 16th Int. Conf. on World Wide Web (WWW)*, pages 697–706. ACM, 2007.

[30] Asuka Sumida and Kentaro Torisawa. Hacking Wikipedia for hyponymy relation acquisition. In *Proc. of IJCNLP 2008*, pages 883–888, 2008.