Wanderlust: Extracting Semantic Relations from Natural Language Text Using Dependency Grammar Patterns

Alan Akbik
Freie Universität Berlin
Institute of Computer Science
14195 Berlin, Germany
akbik@inf.fu-berlin.de

Jürgen Broß
Freie Universität Berlin
Institute of Computer Science
14195 Berlin, Germany
bross@inf.fu-berlin.de

ABSTRACT

A great share of applications in modern information technology can benefit from large coverage, machine accessible knowledge bases. However, the bigger part of todays knowledge is provided in the form of unstructured data, mostly plain text. As an initial step to exploit such data, we present Wanderlust, an algorithm that automatically extracts semantic relations from natural language text. The procedure uses deep linguistic patterns that are defined over the dependency grammar of sentences. Due to its linguistic nature, the method performs in an unsupervised fashion and is not restricted to any specific type of semantic relation. The applicability of the proposed approach is examined in a case study, in which it is put to the task of generating a semantic wiki from the English Wikipedia corpus. We present an exhaustive discussion about the insights obtained from this particular case study including considerations about the generality of the approach.

1. INTRODUCTION

1.1 Motivation

A great share of applications in modern information technology can benefit from large coverage, machine accessible knowledge bases. In the first place this fits to applications conceived by the semantic web community, semantic search being the most prominent example. However, many other application areas can be named that benefit from the availability of structured knowledge. Amongst others these primarily include the application fields of natural language processing (NLP) and information retrieval (IR). WordNet [12] is a perfect a example for a (lexical) knowledge base that is leveraged by NLP applications such as word sense disambiguation [3], machine translation [7] or sentiment analysis [9, 10]. In the field of IR, tasks such as query expansion [19] or question answering [13, 15] are examples that take advantage from access to structured background knowledge.

Existing knowledge bases like WordNet [12], Cyc [18] or SUMO [20] have been manually constructed in a laborious and expensive process. While such a procedure offers very high precision, it naturally does not scale well. High coverage and up-to-dateness are difficult to be achieved this way. Inspired by success stories such as Wikipedia, recent approaches pursue the idea of building large-scale structured knowledge bases by leveraging the collaborative power of thousands of volunteers [26, 22]. We believe this to be a very promising way. However, similar to other Web 2.0-style applications, such an approach depends heavily on a critical mass of users. In fact, this approach entails a circular dependency: A critical mass of users can only be attracted if enough structured data has been collected so that people can start developing compelling applications. But without being able to communicate the benefits (by means of compelling applications), it is hard to convince people to start semantically annotating their data. To overcome this particular dilemma or to generally acquire large-scale semantic knowledge bases, techniques are needed that are able to structure a large amount of existing data without requiring human intervention. In this paper we propose a method that utilizes the dependency-style information provided by a deep grammatical parser. Our hypothesis is that universal grammatical patterns exist which are used to express arbitrary relations between two entities and thus can be exploited in an extraction process. A case study is provided that puts this assumption to the test.

1.2 Related Work

There exists a variety of information extraction approaches that are used to acquire semantic relations from natural language text. We differentiate between methods that address the whole Web as a corpus and techniques that focus on more restricted corpora such as Wikipedia. The latter corpora typically provide a rich set of structured metadata which can be leveraged by extraction procedures. For example in the case of Wikipedia many articles are augmented with so called infoboxes, containing related attribute/value pairs. Furthermore, articles may also be classified into a hierarchical category structure existent in Wikipedia.

A main challenge to extracting information from the Web is its inherent heterogeneity and large scale, thus hindering[1] approaches utilizing deep linguistic analysis. Early systems

SemSearch '09 Madrid, Spain

[1]Parsing natural language raises high costs in terms of time and resource consumption. However, our belief is that in the near future and even today with the advent of cheap and easily accessible compute clusters (keyword "cloud computing", e.g. Amazon EC2) this issue will be of minor importance.

are [1] and [5] which employ a pattern matching approach to derive arbitrary binary relations from web pages. Since both define patterns only over shallow linguistic information, they are exposed to even small variations in the linguistic expression of a relation. Like the majority of systems in that chain of work, they make use of the inherent information redundancy present in the Web which allows for statistical assessment of extracted relations. More recent systems like *KnowItAll* [11] and *TextRunner* for example [4] assess extractions by utilizing the counts provided by search engine hits to compute the relatedness of terms in a particular extraction.

Due to relatively broad coverage and advantageous characteristics (see Section 3.1), Wikipedia has been found to be an invaluable source for knowledge extraction. The majority of systems that utilize this corpus, exclusively make use of the structured metadata available in Wikipedia, but disregard the information hidden in the articles themselves. The approach of *DBPedia* [2] relies on the infoboxes available for many Wikipedia articles. These provide structured information which is consistently-formatted for articles describing instances of the same type. For a predefined subset of article types, handcrafted extractors have been written in *DBPedia*, yielding a multi-million set of RDF triplets. Further semantic annotations that are available in Wikipedia are categories which are arranged in a hierarchical structure. Wikipedia articles may be associated to one or more of these named categories. Ponzetto et al. [21] use a set of shallow linguistic heuristics applied to the category tags in order to infer a taxonomy over the concepts described by Wikipedia articles. The *Yago* system [24] links Wikipedia category labels to WordNet entries. It then exploits the hypernym and hyponym relations available in WordNet to derive a highly accurate taxonomy. To derive further relations *Yago* depends on manually constructed rules applied to the Wikipedia category structure. While [21] only allows to extract taxonomic relations and *Yago* is limited to a predefined set of relations, Wanderlust is capable of extracting arbitrary relations.

Recent work most similar to our approach are [17], [27] and [25]. While previously mentioned approaches exclusively rely on the existence of structured data, these systems allow to extract semantic relations from natural language text. Like Wanderlust, [17, 25] make use of the grammatical structure provided by a deep parse. Nakayama et al. [17] analyze the phrase structure trees produced by the Stanford NLP Parser [16]. Their extraction process is controlled by a small set of handcrafted patterns defined over the phrase structure trees, limiting the coverage of their approach. A heuristic based on link structure analysis of the Wikipedia corpus is used to classify sentences as important for an article. Only these sentences are considered in the extraction process. However, they do not examine the impact of this heuristic. The *Kylin* system [27] exploits the correspondence between article text and structured data of infoboxes. Relations expressed in infoboxes are heuristically matched to sentences in the corresponding article. This way, sentences are labeled as containing specific relations. These labels are then used to learn relation specific extractors. Their approach achieves very high precision, but is limited to semantic relations that exist in infobox templates. Similar to Wanderlust the *Leila* information extraction system [25] uses the linkages produced by the Link Grammar NLP Parser [23]. The system combines deep linguistic analysis with machine learning methods. In the learning phase it relies on the existence of attribute/value pairs describing instances of a specific relation. As such the system is limited to relations for which instances are available in structured format. In this paper we examine which grammatical relations provided by an NLP parser are universally applicable to extract arbitrary relations.

1.3 Outline

The remainder of this paper is organized as follows. In Section 2 we present the general idea of the Wanderlust extraction algorithm, explaining in detail how it utilizes the grammatical dependency information obtained from the Link Grammar Parser. Section 3 gives an overview about the case study we have conducted. We describe how Wanderlust is applied to the Wikipedia corpus in order to populate a semantic wiki system. In Section 4 we present results of our experiments and provide an exhaustive discussion of the insights gained from this case study. The discussion includes considerations about the general applicability of the proposed approach. Section 5 concludes the paper.

2. WANDERLUST

2.1 Problem Statement

We define the problem that is to be solved by the proposed method as follows: Goal is the extraction of *semantic relations* from plain text in a subject-predicate-object triplet form, analogous to statements in RDF[2]. Subjects and objects represent concepts defined by the meaning of their term and are subsequently referred to as *entities* (resources in RDF). *Predicates* are used to describe the nature of the relationship between two entities with a sequence of words. The extraction method is intended to be usable for any kind of text sample containing grammatically correct sentences in English. The extraction of semantics is solely based on grammatical properties of the sentence, as the assumption is that no structured data can be taken into account.

We restrict the problem as follows: Implicitly expressed semantic relations[3] and relations that span more than a single sentence are *not* subject to extraction.

2.2 Proposed Method

2.2.1 Dependency Parse Linkages

The main hypothesis behind the algorithm is that certain grammatical structures exist which are universally valid and therefore allow for the extraction of arbitrary semantics. In order to find a set of grammatical patterns that express relations between entities, a deep linguistic analysis of sentences is performed using the information given by a dependency-style deep grammatical formalism called *link grammar* [23]. In this formalism, links are drawn above grammatically dependent terms within a sentence. These links are labeled according to the nature of the grammatical relationship of

[2]http://www.w3.org/TR/REC-rdf-syntax

[3]Implicitly expressed semantic relations need to be derived from the context and may require common sense knowledge. Consider the following example: "Like Mercedes-Benz, Volkswagen is a German carmaker." Here, it is implicitly stated that Mercedes-Benz is a German carmaker.

two terms. E.g. the link label "**D**" is used to connect a determiner to a noun, while "**S**" is used to connect a subject to a verb. If not directly connected, one of the properties of the formalism (called *connectivity*) ensures that all terms of the sentence are at least indirectly connected via a number of intermediary terms. A path between two words of a sentence is called a *linkpath*. The source and target of such a linkpath are denoted as *start term* and *end term* respectively. The set of all links describes the grammar of the entire sentence and is referred to as *linkage*.

Within this formalism it can be argued that if a direct relationship between two terms is expressed by linking them together, then a *chain of connected terms* describes the relationship between a start and stop term. This can be seen from two perspectives. From a grammatical point of view it can be argued that the sequence of link labels (the linkpath) from start to stop term describes the grammatical relationship between these terms. We argue that certain grammatical relationships between two terms also imply a semantic relationship. Therefore, if a linkpath fulfills necessary grammatical criteria, the sequence of words interlinked on that path (subsequently denoted as *wordpath*) can be seen as describing the semantic relationship between two terms.

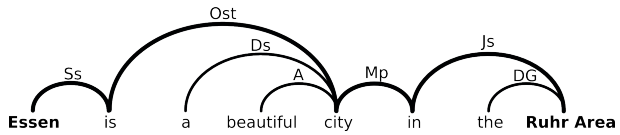

Figure 1: Linkage for an example sentence.

As an example consider the sentence "*Essen is a beautiful city in the Ruhr Area.*". Its linkage is illustrated in Figure 1. The chain of connected terms used in the example is highlighted. The individual parts of the chain are listed below:

1. Start term: "Essen"
2. Stop term: "Ruhr Area"
3. Wordpath: "is" - "city" - "in"
4. Linkpath: {Ss, Ost, Mp, Js}

In this example, the wordpath accurately describes the relation between start and stop term. By combining the terms in the wordpath to form a single predicate, the valid relation IsCityIn(Essen, Ruhr Area) is generated. Refer to Figure 2 for an illustration of this.

Important to note is that a valid relation has been generated without considering the lexical meaning of terms in a sentence. The only information that is used in the relation extraction process is the linkpath, i.e. the grammatical relationships between words in a sentence. If a chain of terms with the same linkpath is found in the linkage of a different sentence, a valid semantic relation can be generated using the same method. To grasp the idea, consider exchanging one or multiple terms in a sentence with terms that are treated equally by the link grammar parser. E.g. replacing "city" with "place", yielding the sentence "*Essen is a beautiful place in the Ruhr Area*", from which using the same linkpath the valid relation IsPlaceIn(Essen, Ruhr Area) is extracted. Thanks to the application of a deep linguistic parser, sentences may also differ more strongly (i.e.

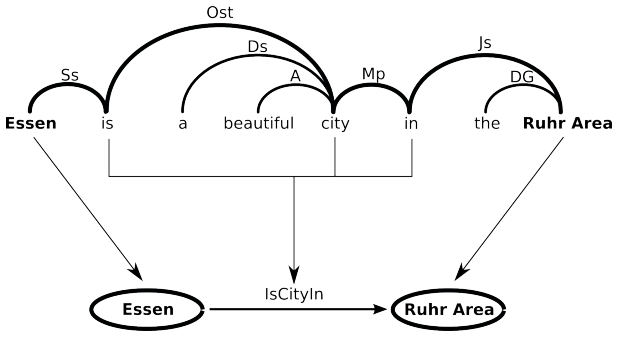

Figure 2: Relation extraction from an example sentence.

inserted relative phrases, additional modifiers). As long as two terms are connected with a valid linkpath, the algorithm is able to find a suitable predicate to express their semantic relationship.

However, a great share of observable linkpaths is not useful with regard to the preceding considerations. To illustrate this, consider two arbitrary words in the example sentence, such as "beautiful" and "Ruhr Area". The wordpath connecting "beautiful" to "Ruhr Area" is "CityIn". A relation built using this information would be CityIn(beautiful, Ruhr Area), which is nonsensical and therefore false. The problem here is not that the original theory is incorrect, but rather that the sentence does not explicitly state any information which connects both terms in question. The adjective "beautiful" modifies another noun, but not "Ruhr Area". Note that because of this even a human annotator would have difficulties finding a predicate to connect "beautiful" to "Ruhr Area" in a way that expresses the semantics of the sentence.

A major challenge is to identify a set of valid linkpaths that can be used for the extraction of semantic relations.

2.2.2 *Valid Linkpaths*

By initial observation some linkpaths were found to generally represent usable wordpaths, while many others were not. The task of finding a set of valid linkpaths was solved by manually annotating a training set with relation triplets. Wanderlust was then applied to the annotated corpus, using any possible linkpath to obtain relation triplets. Upon completion, for each distinct linkpath the number of true positives were counted and divided by the total number of positives, establishing a coefficient representing the level of confidence the algorithm has when extracting relations using the corresponding linkpath.

The coefficients for the linkpaths were generated from an annotated corpus of 10,000 sentences, resulting in a total of 46 valid linkpaths. Examples from the most common linkpaths and the type of predicate they find are listed in Table 1. The table shows how grammatically distinct types of predicates can be found using the Wanderlust mechanism. Linkpaths such as 1, 3 and 4 use the information given by two terms that function as subject and object to a verb (or verbal expression in the case of 3) which can therefore be used as predicate for the terms. Another case are linkpaths such as 2 (used in the example of Section 2.2.1), in which two nouns are connected with a predicate incorporating a third noun. Linkpath 5 is an example of a more rare construct

in which the predicate consist of one verb connected to the infinitive form of another.

Table 1: Examples for the most common linkpaths.

#	Linkpath	Example predicate
1	Ss Ost	Is
2	Ss Ost Mp Js	IsCityIn
3	Ss Pv MVp Jp	WasKilledBy
4	Ss PP Os	HadCaptured
5	Ss TO I Os	FailedToDefeat

Using the 46 linkpath that have been identified, Wanderlust is able to extract relation triplets from link grammar linkages. The ensuing section illustrates this process applied to a specific use case.

3. CASE STUDY

In the previous section it has been stated that the major hypothesis underlying the idea of the Wanderlust algorithm is the existence of a set of grammatical patterns that express relations between entities. In this section we give an overview about a case study we conducted putting this hypothesis to the test. We apply Wanderlust to the English Wikipedia corpus and use the obtained semantic relations to populate a semantic wiki. Besides having the positive side-effect of bootstrapping a semantic wiki, the main goal of the case study is twofold. On the one hand we are interested in deriving quantitative results about the performance of Wanderlust on a specific corpus. On the other hand a major intent is to gain general qualitative insights about the applicability of Wanderlust's linguistic approach.

3.1 The Case for a Semantic Wikipedia

In this section we outline why we choose to "semantify" the Wikipedia corpus in our case study. We further provide some information about the semantic wiki platform we employed to capture the extracted semantic relations.

Numerous characteristics make the Wikipedia Online Encyclopedia[4] an attractive corpus for the extraction of semantic relations:

Coverage and Diversity: To date the English Wikipedia contains more than 2.7 million articles about concepts from all aspects of life. It represents the most complete encyclopedia worldwide and is thus an invaluable source for knowledge extraction.

High Quality of Content: The results of several studies e.g. [14] indicate a very high quality-of-content for Wikipedia.

Actuality: Thanks to the openness and the collaborative approach the knowledge contained in Wikipedia is less exposed to ageing.

Factual Language: Due to the encyclopedic nature and because of its large and active community the quality of writing (in terms of use of proper grammar and spelling) is high and factual. This greatly facilitates any computerlinguistic approach.

Internal Link Structure: Each concept defined by an encyclopedic entry in Wikipedia has its own unique URI. When mentioning a specific concept in an article, authors are encouraged to insert a reference (so called *page links*) to the corresponding URI. These references allow to find named entities in unstructured text. Furthermore thanks to unique URIs word sense disambiguation can easily be performed.

In our case study we employ the Semantic MediaWiki (SMW) system [26] to store the results of the extraction process. SMW is a natural extension to the MediaWiki software that also powers the Wikipedia platform. The goal of the SMW developers is to add a specific layer of machine-readable metadata to a wiki, thus enabling semantic web applications. Their ultimate vision is to integrate the extensions into Wikipedia allowing for a large-scale collaborative approach to construct an open semantic knowledge base.

As principal extension to the standard MediaWiki software they introduce the concept of *typed links*. The type of a link is given by sequence of words describing the nature of a relationship between articles. E.g. the link between the articles *Berlin* and *Germany* may be assigned the type *isCapitalOf*. No restrictions are imposed on the choice of the value of a link type. This small extension allows for a similar knowledge representation and semantic expressiveness such as RDF triples. *Subject* and *object* of such a triple are concepts defined by articles which in turn are identified by unique URIs. The *predicate* is given by the type of the link connecting both articles. In fact, the SMW software allows its contents to be exported into the RDF format.

The formal knowledge model of SMW is augmented by the introduction of so called *subproperties*. This extension allows to define one link type (i.e. predicate) to be the *subtype* of another. E.g. the predicate *IsGoodFriendOf* may be defined as subtype of the predicate *IsFriendOf*. The subproperty concept can be seen as introducing the (meta) relation *SubtypeOf* that takes as arguments two predicates. This way predicates are ordered along a chain of implication, i.e. one predicate implies all of its subtypes. Take note that this layer of logic allows for more sophisticated querying of the underlying knowledge base. For instance a query over the predicate *IsFriendOf* will also consider the predicates *IsGoodFriendOf* and *IsVeryGoodFriendOf*. In Section 3.2.4 we show how Wanderlust can be extended to capture the *SubtypeOf* relation.

3.2 Wanderlust Applied

This section illustrates the use of Wanderlust set to the specific task of generating a semantic wiki using the English Wikipedia corpus. A number of extensions made to the algorithm either out of necessity or expedience are discussed. Refer to Figure 3 for an outline of the entire algorithm.

The algorithm is applied to the English Wikipedia corpus dated from October 2008 which contains slightly more than 2.4 million entries. Due to the CPU intensive natural language parsing task, the algorithm is run in parallel on a cluster of 50 commodity hardware machines. Parsing results are stored in a separate database. Wikipedia articles are analyzed one by one. A preprocessor removes all markup and meta information from the page, passing page links and terms written in bold face to a procedure called *entity tagger*. It finds named entities in an article and in a subsequent step disambiguates them. The output of the

[4]http://www.wikipedia.org

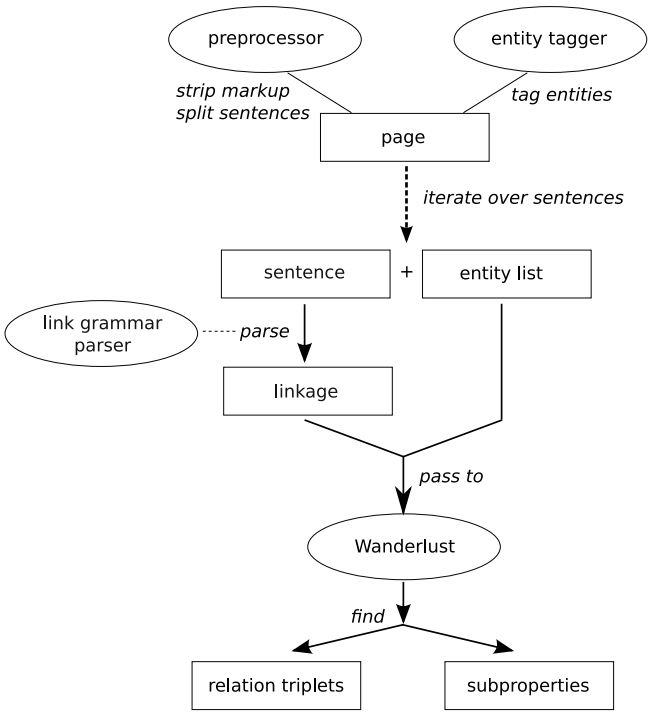

Figure 3: Outline of the steps used in the process to extract semantic relations from Wikipedia articles.

procedure is a list of disambiguated entities (denoted as *entity list* in the following) that are valid for the article. A precise description of this method is given in Section 3.2.1.

The preprocessor proceeds to split the page into a set of sentences. For each sentence the algorithm finds all terms that refer to an entry in the extracted entity list. All sentences that contain at least two entities are passed to the link grammar parser, while all others are dismissed. To enhance the performance of the parser, sentences are rewritten beforehand as described in Section 3.2.2. The obtained linkages are input to Wanderlust which attempts to extract semantic relations for all pairs of contained entities. The output of Wanderlust are relation triplets which are stored in the SMW database.

In addition to this, the algorithm also attempts to identify *SubtypeOf* relations (see Section 3.1) between extracted predicates. The exact procedure is outlined in Section 3.2.4.

3.2.1 Entity Recognition

As mentioned in the algorithm outline, as a first step named entities need to be extracted and disambiguated to corresponding Wikipedia pages. The heuristic used to achieve this is based on information provided by page links and Wikipedia's conventions regarding synonyms.

The underlying idea is that a page link will link a term to the Wikipedia page that provides disambiguation according to the context in which the term is stated. E.g. in the sentence "Apollo killed Python" a page link set for the term "Python" will lead to the page "Python_(mythology)", therefore providing correct disambiguation. We take this idea one step further and reason that given no other page links with anchor text "Python" within the same page, all occurrences of the term refer to the same word sense. This means that by virtue of page link information the algorithm can compile a list of terms to which Wikipedia page titles are assigned. Any occurrences of these terms within the page can then be tagged as entities. If multiple page links with the same anchor text but different link targets have been found in an article, the heuristic cannot be applied for this specific term.

The topic of an article itself is handled differently. Very few pages contain page links to themselves, meaning that the term describing the topic of the page (*page term*) cannot be found using the method introduced above. Instead the page title is analyzed, which in many cases directly matches the primary page term. Page titles can therefore be used as page terms with the following exceptions:

Since page titles are unique, pages for ambiguous terms have different titles for each distinct meaning. Several conventions in Wikipedia exist that describe how to choose appropriate page titles. One convention is to use the page term with context information written in brackets as page title. An example of this is "Python_(mythology)". Other conventions for instance affect the disambiguation of places, e.g. "Berlin, Pennsylvania". Simple heuristics that strip of parts of the page title can be applied to obtain a reasonable page term.

Another helpful style convention in Wikipedia is to write the page term (including all synonyms) in bold face in the first paragraph of a page. By convention, this should be the only use of bold face within the entire page. This observation allows to easily find synonyms of the page term and thus to expand the list of disambiguated named entities.

However, not all interesting terms have page links. The ratio of page links per sentence has been found to vary wildly within the English Wikipedia corpus. Because sentences need at least two entities in order to be usable for Wanderlust, an additional enrichment technique was applied. All terms which are composed of more than one word (such as "Federal Republic of Germany") were blindly assigned a Wikipedia page with an identical page title provided it exists. The reasoning behind this step is that concepts consisting of more than one word are less likely to be ambiguous than one word concepts.

3.2.2 Sentence Modification

After the tagging of entities, sentences are modified. All entities which consist of more than one word are written together, so that the parser will treat the entire entity as one word. As this form will in most cases not be part of the dictionary of the link parser, it will guess the word type of the unknown word to be a proper noun.

The motivation for this is that while in many cases the parser correctly identifies subsequent nouns to be part of the same entity (connected by the G or GN link in a linkage), it has problems with entities which include words other than nouns. Examples for this are "Federal Republic of Germany" or "Johann von Goethe". Such entities are not treated as a noun but rather as two nouns connected by a preposition. We found that this incorrect treatment of entities has adverse effects on the quality of the parse. By writing all entities as one word, this problem is avoided. See Figure 4 for two linkages of the example sentence "*Frederick I became the elector of the Margraviate of Brandenburg*". In the first linkage, the sentence is parsed in its original syntax, while in the second the two entities "Frederick I" and "Margraviate of Brandenburg" are each written as one word.

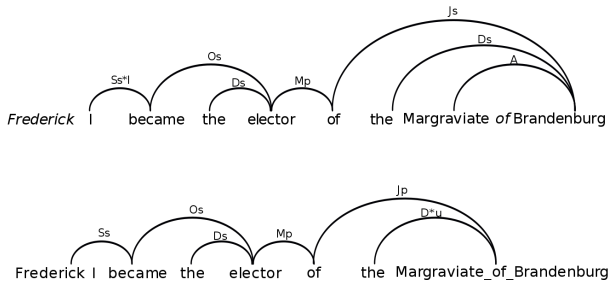

Figure 4: Example of sentence modification and the resulting linkages.

In the first linkage, the parser could only parse the sentence by skipping two words. The resulting linkage describes the semantically very different sentence "*I became the elector of the Margraviate Brandenburg*". The modified sentence is less complex and as a result correctly parsed.

3.2.3 Expanded Wordpaths

When a noun is part of the wordpath, e.g. the sequence of words describing the semantic relationship between subject and object, all its modifying terms such as adjectives and number words should be considered. Terms that modify the meaning of the noun need to be included in the wordpath in order for it to more accurately mirror the semantics stated within a sentence. A noun with all modifiers is denoted as *expanded noun*.

This is best illustrated with an example sentence, such as "*Krypton is a fictional planet in the DC_Comics_Universe*". A linkage for this sentence is illustrated in Figure 5. In this linkage, the relation IsPlanetIn (Krypton_(Comics), DC_Comics_Universe) is found with the predicate containing the noun "planet". While this relation fits the semantics of the sentence adequately, a more precise predicate is possible using the expanded noun "fictional planet": By incorporating the expanded noun instead of the regular noun into the predicate, the relation *IsFictionalPlanetIn* (Krypton_(Comics), DC_Comics_Universe) is extracted. An illustration of the example is given in Figure 5.

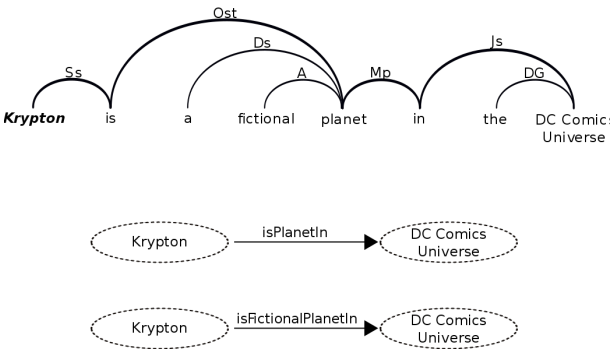

Figure 5: Linkage for the example sentence. The upper relation is extracted if expanded noun information is disregarded. The lower relation is extracted using the expanded noun.

A wordpath using expanded nouns is referred to as *expanded wordpath*. It allows for more accurate representation of a sentence's semantics.

3.2.4 Subtypes

As mentioned in Section 3.1, the formal knowledge model of SMW introduces the possibility to express subtype hierarchies of predicates. In the following we describe in which way the identification of expanded wordpaths enables the automatic generation of such predicate subtypes.

If an expanded noun is part of a wordpath, the noun's modifiers are dropped one by one. With each discarded modifier, the wordpath becomes less specific. The wordpath including the dropped modifier is saved as subtype to the wordpath without. This way the algorithm infers a chain of implication.

Consider the example sentence "*Dirk is a very good friend of Elmar*". The wordpath from "Dirk" to "Elmar" is IsFriendOf. The expanded wordpath is IsVeryGoodFriendOf, containing the two noun modifiers "very" and "good". The former is dropped first, yielding the predicate IsGoodFriendOf to which IsVeryGoodFriendOf is a subtype. In a second step, the term "good" is also dropped, yielding IsFriendOf to which IsGoodFriendOf is a subtype.

Within the case study, using this approach a total of 749,703 subtypes are identified. Furthermore, Wanderlust generates over 2.5 million relation triplets from the English Wikipedia corpus encompassing 312,744 relation types. More detailed results are presented in the ensuing section.

4. RESULTS AND DISCUSSION

The stated goal of the algorithm is to extract arbitrary relation triplets from plain text in order to make its content available for semantic applications, such as semantic search. The algorithm attempts to generate a knowledge base consisting of all information communicated by the analyzed sentences. In this section we discuss the results obtained by the conducted case study. We analyze the quality of the knowledge model generated by the application of Wanderlust to the English Wikipedia. Strengths and weaknesses of the proposed method are illustrated.

4.1 Extracted Relation Types

Because the algorithm can generate arbitrary predicates, the number of distinct relation types can potentially be very high. Indeed, in the result set a total of 312,744 distinct predicates are found. The most common relation types are listed in Table 2. Most of the predicates convey either taxonomic relations or give information concerning persons, institutions or locations, which reflects the encyclopedic nature of Wikipedia.

The high number of relation types has both positive and negative aspects. On the plus side the use of many distinct relation types allows for the correct and unabstracted modeling of knowledge, making it possible to fit relation triplets precisely to the information stated in a sentence. On the other hand, processing this information in semantic applications becomes more difficult since it is more feasible to define a layer of logic for a knowledge base with a limited set of predicates.

The two sides of this problem can be illustrated with the application scenario semantic querying. High numbers of predicates allow the user to pose very specific queries. Problems arise for synonymous or near-synonymous predicates, allowing facts to be expressed in a multitude of ways. A

Table 2: List of most common predicates.

Predicate	# Relations	Category
Is	627357	Taxonomic
Was	203237	Taxonomic
WasBornIn	50234	People
Has	25082	
Became	17479	Taxonomic
IsLocatedIn	14264	Location
IsVillageIn	13854	Location
IsTownIn	12464	Location
WasFoundedIn	11184	Institution
Won	10192	Institution / People
Attended	10046	People
Had	9281	
IsCityIn	8899	Location
Joined	8719	People
IsSpeciesOf	8702	Taxonomy / Biology
IsTributaryOf	7038	Location
IsGenusOf	6900	Taxonomy / Biology
IsCommuneIn	6350	Location
Defeated	6347	People / Institution
Played	6226	People
IsNameOf	5910	
IsTributaryIn	5771	Location
WasEstablishedIn	5594	Institution
IsVillageOf	5526	Location
WasElectedTo	5507	People
IsMemberOf	5403	Institution / People
Received	5371	
WasElectedIn	5126	People
Released	5068	

user striving to answer the query WasKingOf(?, England) can only obtain a full list from the result set if the query is posed with various predicates, such as TookThroneOf(?, England), BecameKingOf(?, England), etc. In SMW, this problem can be compensated by defining subproperties, of which 749,703 were generated by Wanderlust and added to the knowledge base.

4.2 Accuracy of Extraction

In order to obtain quantitative values on the performance of Wanderlust it was decided to manually annotate a number of random Wikipedia pages with all data that a human reader could find. A total of 4005 sentences were annotated containing 1278 relations. This is a ratio of approximately one relation every three sentences which might appear scant. The reason for this is that data was only annotated if it could be put in the form of a subject-predicate-object triplet with the restriction of using a "reasonable" number of words (5) in a predicate and only entities with *existing* pages in the corpus. Adhering to the the problem statement (Section 2.1), only information that is explicitly stated within one sentence is used. The annotated pages constitute the gold standard for the quantitative analysis.

Applying Wanderlust on the annotated corpus and comparing the results with the gold standard, a total of 206 true positives were found, putting **raw recall at 16.1%** compared to what a human reader would find within the text. With 206 true out of 251 total positives, this puts the algorithm's **raw precision at 82.1%**. The semantic relations found by Wanderlust were classified into several categories representing the impact they have on the generated knowledge base. An overview of the distribution is given in Table 3.

Table 3: Correctness of relations.

Group	#
Useful	155
Unhelpful	51
Nonsensical	24
Untrue	21

The most common class of relations found is *useful* true positives which positively contribute to the model of knowledge built by Wanderlust. Less useful, but not false, are relations of class *unhelpful*. These relations, such as Has(Moon, Density) or Is(Kármán_line, Definition) are semantically to weak in order to be useful for most application areas. Relations of class *nonsensical* are meaningless statements such as IsPlanetFrom(Jupiter, Planet) or WasObservedTo(Venus, Planet). This information is neither true nor false. While counted as false positives in the above calculation of precision, it can be argued that some application domains such as semantic search are not negatively affected by nonsensical relations, considering that a user will typically not enter nonsensical queries such as WasObservedTo(?, Planet). If nonsensical false positives are taken out of the calculation an overall **precision value of 91.6%** is found. The final class of relations are those which convey *untrue* information and therefore negatively affect the model of knowledge.

In Section 4.3, a number of difficulties which lead to recall and precision loss are analyzed.

4.3 Analysis of Error Sources

This section analyzes and quantifies reasons for recall and precision loss. After carefully analyzing the evaluation set and the data returned by Wanderlust, the following list of error sources was compiled: *Coreferences*, *parse errors*, *entity recognition errors*, *context errors* and *incomplete object errors*. Of all error classes, only incomplete object errors are directly related to Wanderlust, all others being general problems that can be addressed separately.

For an overview of the effect of errors classes on recall refer to Table 4. Missing coreference resolution is responsible for the greatest part of recall loss in the case study. Insufficient tagging of entities is at the second place, accounting for 18.9% of recall loss. In 9.5% of cases, sentences were not sufficiently understood by the link grammar parser. *Other* includes mistakes made in the preprocessing step, e.g. by the sentence splitter. Finally, the incomplete object error is responsible for 12.5%.

Reasons for and quantification of precision loss are listed in Table 5. Because relations of type unhelpful are undesired, the table includes unhelpful relations as false positives. Context errors are responsible for the largest amount of false positives, followed by parse errors and coreference errors. Incomplete object errors account for 13.5% of false positives.

In the following, selected error classes are discussed in more detail with respect to their impact on precision and recall. All errors share the property that they cause valid linkpaths to yield false relations (and vice versa) and are

Table 4: Quantification of recall loss.

	# Relations	Percent
All Relations	1278	100%
Coreference	518	40.5%
Entity error	242	18.9%
Incomplete object error	160	12.5%
Parse error	121	9.5%
Other	31	2.4%
True positives	206	16.1 %

Table 5: Quantification of precision loss.

Class	# False positives	Percent
Context errors	32	33.3%
Parse errors	22	22.9%
Coreference errors	17	17.7%
Incomplete object errors	13	13.5%
Entity errors	12	12.5%

therefore levels of distortion to the algorithm. The incomplete object error is a weakness in the algorithm and compromises the original assumption of the proposed method in some respect.

4.3.1 Coreferences

Coreferences in linguistics are multiple references to the same referent (or entity as in the sense of this project) using different terms. Coreferences may be synonyms, but also personal pronouns ("he", "it"), demonstrative pronouns ("this", "that") or more general terms for a specific entity.

Problems for Wanderlust arise when a relation triplet is extracted for a term which is actually a reference to another entity. In the sentence "The city lies at the river Spree" the term "city" actually references another term. Because coreference resolution is not performed in Wanderlust, the algorithm is unaware of this. To minimize errors made because of this, the algorithm is only permitted to use proper nouns as subjects in relation triplets, the reason being that proper nouns are unlikely to be coreferences of other entities. This restriction taken together with the missing coreference resolution is responsible for a substantial recall loss of the algorithm, accounting for over 40% of false negatives.

Even with the above mentioned measure, coreferences still cause 17.7% of the total precision loss. One problem here is Wanderlust's handling of non-proper noun page terms. All page terms can be used as subjects in relation triplets following the reasoning that a page term will not be used as coreference to another entity within its own page. This however has been found not to be universally true.

4.3.2 Parse Errors

Another problem is the degree of distortion caused by errors in the link grammar parser. The link grammar parser has quality-of-parse variables which it uses to "gauge" the accuracy of a linkage. One important variable is *skip*, which counts the number of words the parser needed to skip in order to parse the sentence. Note that the sentence is parsed as though the skipped words are not part of it. The variable *linkage number* represents the ordering of all possible linkages the parser returns for a given sentence. The lowest linkage number is by estimation of the parser the best linkage. Both variables were kept low in order to to assure a certain linkage quality for Wanderlust.

This means that sentences with more than one skipped word were dismissed from consideration, leading to a recall loss of 9.5%. Sentences which have been incorrectly parsed without having skipped words are especially problematic.

4.3.3 Entity Recognition Errors

The process of entity recognition is an important precursor within use case of applying Wanderlust to populate the Semantic MediaWiki database. Any entity that the entity recognition process fails to tag results in recall loss. Since the tagger is highly dependent on page links in order to disambiguate terms within a page, results vary from page to page. The overall recall loss because of untagged entities has been found to be 18.9%. Wrongly tagged entities make up 12.5% of the overall precision loss. Erroneous tagging can result from one of the basic considerations behind the entity tagger being false, namely that in some cases a term is not page linked to its disambiguated word sense. Another source of error within this context is the fact that falsely or incompletely set page links are existent in the Wikipedia corpus.

4.3.4 Context Errors

Context errors occur if a stated fact is put into context by another part of the sentence or page. In such cases, extracting this fact in form of a relation triplet may yield a false positive. Generally problematic are pages or paragraphs which describe fictional content, express supposition or opinions. The distinction whether a sentence states fact or fiction is not made by Wanderlust. This leads to many errors in pages which describe scenarios or the storyline of books or movies for example. All stated facts within are fiction, but Wanderlust adds them to the model of knowledge nonetheless.

The problem however is not limited to the distinction between fact and fiction. Even factual information can be true only in a certain context. In the sentence "*At night, all cats are gray*" Wanderlust will find Is(Cat, Gray) which is untrue outside of the context "*at night*". Also problematic are sentence like "*It was believed that the earth was flat*" in which Wanderlust will find the untrue Was(Earth, Flat). A sentence with an identical linkage such as "*It is known that the earth is round*" will yield the true relation Is(Earth, Round).

Context errors are not detected by Wanderlust and account for the largest part of precision loss.

4.3.5 Incomplete Object Errors

The initial theory of the proposed method is that certain linkpaths are generally usable to express semantic relations between entities. While many reasons for errors are listed in this section, none have yet challenged this theory. The problem of correctly identifying entities and coreferences is a precursor to, but not part of, Wanderlust.

The error class discussed in this section however directly affects the algorithm's performance. The error occurs when a rule is applied to a verb or an expression which needs more objects than the rule provides in order to be helpful. In many ways it is linked to the theory of *verb valency*. Valency in linguistics is a term which is used to describe how many arguments (i.e. subjects and objects) a verb requires

or "binds" to itself. Depending on the verb and its disambiguated word sense, a verb typically requires between 0 and 2 arguments. Important here is that each verb has a minimum number of objects which are required for it to make sense. In addition to this, verbs may have any number of optional objects.

The verb "to give something to someone" has two necessary objects, but may also bind additional objects to itself for time and place, meaning that is also makes sense as "to give something to someone at some place at some time". A necessary objects cannot be dropped and replaced by one of the optional objects, such as in "to give something at some time", which makes little sense.

The information on which arguments of a verb are necessary and which optional is however *not* reflected in a link grammar parse. Without this information, Wanderlust does not know which paths contain the necessary arguments for a verb, resulting in errors as shown in the following example. In the sentence "*Zeus gave flowers to Hera in Athens*" the verb "to give something to someone" is used which has two necessary objects and one optional (namely "in Athens"). Since Wanderlust does not know which objects are necessary and optional it will find the following relations (classified according to correctness):

1. {Zeus, gave, flowers} - unhelpful, uses only two objects
2. {Zeus, gaveFlowersTo, Hera} - useful, uses the three necessary objects
3. {Zeus, gaveFlowersIn, Athens} - nonsensical, uses three objects, but one optional object instead of a necessary one
4. {Zeus, gaveFlowersToHeraIn, Athens} - useful, uses all three necessary objects plus one

These examples show how the algorithm's unawareness of verb valency results in precision loss. It also has adverse effects on recall because some linkpaths are more sensitive to errors made because of verb valency than others and were dismissed from the list of good linkpaths even though being grammatically valid.

This problem cannot easily be resolved. Even if lists of verbs and their valency were to be obtained from WordNet or a similar resource, it would still have to be disambiguated which meaning of a verb is meant in the context of the sentence. The problem is that ambiguities of differing valency exist for many verbs. The problem is unhandled in the current version of the algorithm and represents a priority for further work.

4.4 General Applicability of the Approach

Some of the aspects discussed so far in this chapter relate directly to the case study we have conducted and therefore are, to some extent, tied to the specifics of the Wikipedia corpus. In this section we discuss the general applicability of the approach, i.e. its extensibility to other domains than Wikipedia or to an open domain scenario respectively. In Section 3.1 we have outlined some advantageous characteristics of Wikipedia including the *coverage and diversity*, the *high quality of content*, the *actuality*, the *factual language* and the *internal link structure*. The first four properties have an implicit influence on the performance of Wanderlust. It is only the last mentioned property which is directly exploited within the case study, i.e. the only real technical dependency is due to the fact that Wikipedia pages and page links are used for entity recognition and entity disambiguation. While the existence of these properties makes life easier in our setting, their non-existence do not hinder the application of our approach.

It is true that the recognition of entities is an important precondition to the Wanderlust algorithm, but this can also be achieved by employing standard techniques and tools for named entity recognition (NER) - the Stanford NER System[5] and the Open Calais Service[6] being the most prominent ones. While entity disambiguation is deemed to be a minor problem in a closed domain, it is a major issue in an open domain setting. Although in a domain other than Wikipedia the internal linkstructure cannot be directly utilized, the information contained in Wikipedia (or other encyclopedic knowledge bases) can be used indirectly to perform entity disambiguation. For example [6] and [8] define a context of a named entity based on encyclopedic knowledge. Entity disambiguation is then performed by computing the contextual similarity of two named entities.

As mentioned above the use of factual language has an implicit (positive) influence on the extraction process. Firstly, factual language tends to express relationships between entities explicitly. Secondly, authors pay attention to using correct English grammar, which is a precondition for useful results of the natural language parser. Employing natural language processing techniques in domains where authors have a tendency to neglect the use of correct English (e.g. Web 2.0 - user generated content) requires preprocessing steps like text normalization.

In summary, our belief is that the presented approach is generally applicable and is not restricted to the Wikipedia domain. The hypothesis underlying the approach is a purely linguistic one and therefore is only bound to English language. Prerequisites such as NER and entity disambiguation can be met by standard techniques as outlined above.

5. CONCLUSION

In this paper we have presented an algorithm based on the hypothesis that universally valid grammatical patterns exist which can be used to extract explicitly stated facts from sentences in plain text. With the use case of populating a semantic wiki we have put this hypothesis to a thourough test and identified challenges both general and specific to the algorithm. The algorithm is able to find a wide variety of distinct relation types using the 46 patterns we have identified. In the use case, a precision value of over 80% was measured, which supports the initial theory.

We have named and quantified the difficulties encountered during the case study and discussed their impact on relation acquisition. While most (coreferences, entity recognition) are not specific to the underlying theory of the algorithm, the problem of incomplete objects contradicts the assumption that linkpaths alone would be enough to model universally valid patterns. Instead, it has been shown that the valency of verbs and verbal expressions must be taken into account. Future work will therefore focus on extending the algorithm to handle the incomplete object error. Given that this can be accomplished, we believe that the original in-

[5]http://nlp.stanford.edu/ner/index.shtml
[6]http://www.opencalais.com/

tention of the proposed method can be achieved. Another direction in future work is to investigate methods that allow to identify clusters of semantically identical or near similar relations.

Take note that the core extraction algorithm can naturally be complemented by statistical assessment methods that utilize information redundancy available in larger corpora.

6. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: extracting relations from large plain-text collections. *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, 2000.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *Lecture Notes in Computer Science*, 4825:722, 2007.

[3] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. *Lecture notes in computer science*, pages 136–145, 2002.

[4] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. *Proceedings of IJCAI*, 2007.

[5] S. Brin. Extracting patterns and relations from the world wide web. *Lecture Notes in Computer Science*, pages 172–183, 1999.

[6] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, 2006.

[7] N. Chatterjee, S. Goyal, and A. Naithani. Resolving pattern ambiguity for english to hindi machine translation using wordnet. *Workshop on Modern Approaches in Translation Technologies*, 2005.

[8] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. *EMNLP 2007: Empirical Methods in Natural Language Processing*, 2007.

[9] X. Ding, B. Liu, and P. Yu. A holistic lexicon-based approach to opinion mining. *Proceedings of the international conference on Web search and web data mining*, pages 231–240, 2008.

[10] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. *Proceedings of the International Conference on Language Resources and Evaluation*, 6, 2006.

[11] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM New York, NY, USA, 2004.

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

[13] A. Frank, H. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer. Question answering from structured knowledge sources. *Journal of Applied Logic*, 5(1):20–48, 2007.

[14] J. Giles. Special report–internet encyclopaedias go head to head. *Nature*, 438(15):900–901, 2005.

[15] G. Kasneci, F. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. Naga: Searching and ranking knowledge. *IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008*, pages 953–962, 2008.

[16] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430. Association for Computational Linguistics Morristown, NJ, USA, 2003.

[17] T. H. Kotaro Nakayama and S. Nishio. Wikipedia link structure and text mining for semantic relation extraction. In *Proceedings of the Workshop on Semantic Search at ESCW*, volume CEUR Workshop Proceedings, pages 59–73, Tenerife, Spain, June 2008.

[18] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49, 2006.

[19] D. Milne, I. Witten, and D. Nichols. A knowledge-based search engine powered by wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 445–454. ACM New York, NY, USA, 2007.

[20] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9. ACM New York, NY, USA, 2001.

[21] S. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the national conference on artificial intelligence*, volume 22, page 1440, 2007.

[22] S. Schaffert. Ikewiki: A semantic wiki for collaborative knowledge management. In *1st International Workshop on Semantic Technologies in Collaborative Applications (STICA'06)*, Manchester, UK, 2006.

[23] D. Sleator and D. Temperley. Parsing english with a link grammar. In *Third International Workshop on Parsing Technologies*, 1993.

[24] F. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[25] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717, 2006.

[26] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. *Proceedings of the 15th international conference on World Wide Web*, pages 585–594, 2006.

[27] F. Wu and D. Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth Conference on information and knowledge management*, pages 41–50. ACM New York, NY, USA, 2007.