Bootstrapping Semantic Lexicons for Technical Domains

Patrick Ziering[1] **Lonneke van der Plas**[1] **Hinrich Schütze**[2]

[1]Institute for NLP, University of Stuttgart, Germany
[2]CIS, University of Munich, Germany
{Patrick.Ziering, Lonneke.vanderPlas}@ims.uni-stuttgart.de

Abstract

We address the task of bootstrapping a semantic lexicon from a list of seed terms and a large corpus. By restricting to a small subset of semantically strong patterns, i.e., coordinations, we improve results significantly. We show that the restriction to coordinations has several additional benefits, such as improved extraction of multiword expressions, and the possibility to scale up previous efforts.

1 Introduction

High-quality semantic lexicons are needed for many natural language processing (NLP) tasks like information extraction and discourse processing (Riloff and Shepherd, 1997). Building such lexicons manually is costly and time-consuming. Automatic lexicon construction is therefore an important task and much prior work has addressed it.

This paper adopts *Basilisk* (Thelen and Riloff, 2002) as its basic approach, a system that uses lexico-syntactic patterns for bootstrapping. We have adapted Basilisk to our setting in several ways. Whereas the original Basilisk covers a wide variety of lexico-syntactic patterns, we restrict ourselves to a specific type of patterns, i.e., coordinations. Coordinations have been exploited in lexical acquisition before (e.g., Roark and Charniak (1998); Caraballo (1999); Goyal et al. (2010)). Most of this previous work uses a pairwise perspective (i.e., a focus on whether two words cooccur in a coordination). However, we use coordinations in a Basilisk approach, for which patterns contain several terms in general. We therefore do not split up coordinations in pairs but keep the complete coordination intact. Coordinations in technical domains frequently contain more than 2 elements.

We argue that bootstrapping methods, known to be particularly sensitive to the ambiguity of terms and contexts and prone to semantic drift, benefit from the strong semantic coherence found in coordinations. The elements of a coordination often have a common hypernym; i.e., they are co-hyponyms or members of a common semantic class.

Furthermore, the high arity of coordinations, the fact that coordinations have two or more arguments (e.g., "platinum, nickel, palladium, copper, silver, or gold") further constrains the semantics. For example, if two out of three elements in a coordination have already been identified as substances, this makes it likely that the third is also a substance. General or lexico-syntactic patterns in lexical bootstrapping have arity 1, i.e., they have a single argument. These patterns are too often not restrictive enough to prevent false terms infecting the semantic lexicon, which can lead to semantic drift. For example, given a corpus in which the semantic class DISEASE is not predominant, after some iterations the weak pattern "treatment of <X>" is selected, which also provides many off-class candidates (e.g., treatment of *prisoners*). In coordinations, the semantic coherence among terms leads to more selective patterns. For example, the coordination "congenital heart defect, atherosclerosis, <X>, scleroderma or tuberous sclerosis" can only hold a slot for diseases.

We will show that the restriction to coordination patterns has several additional benefits. The focus on simple coordination patterns circumvents the need for identifying various syntactic relations (e.g., subject), that are part of the extraction patterns of the original Basilisk. Therefore, it circumvents the need for parsing. Syntax in the patent domain, we address in this paper, is complex and characterized by long clauses (cf. average number of tokens per sentence: BNC: 19.7; Brown: 21.3; WSJ: 22.4; Wikipedia: 24.3; EPO: 32.4). The shallow parser used by Thelen and Riloff (2002) would need several months to parse our corpus.

International Joint Conference on Natural Language Processing, pages 1321–1329,
Nagoya, Japan, 14-18 October 2013.

Furthermore, the restriction to a subset of very strong bootstrapping patterns limits the overproduction of patterns radically. We can therefore apply our method to the largest domain-specific corpus that has been used for semantic bootstrapping so far.

Lastly, we benefit from the increased precision in extracting multiword expressions (MWEs) when using coordination patterns. In contrast to original Basilisk and most prior work, we learn terms of any length, because, as we will see later, in the technical domain under consideration MWEs are predominant (e.g., “alkyl trimethyl ammonium methosulfate”).

The paper is structured as follows. In Section 2, we describe our data set, the task we address and our evaluation methodology. Section 3 describes Basilisk and our adaptations, in particular, the context patterns we use. Experimental setup and results are presented in Section 4. In Section 5, we discuss and analyze these results. The last two sections describe related work and conclusions, respectively.

2 Data, task description and evaluation methodology

Data. We use the patent data distributed by the European Patent Office[1] (EPO) as our corpus. We extract the description (the main part of a patent) from 561,676 English patents filed between 1998 and 2008 and perform sentence splitting and tokenization using Treetagger (Schmid, 1994) and lemmatization and part-of-speech (POS) tagging using MATE (Bohnet, 2010). Sentences up to a size of 100 tokens extracted from a sample of 25,000 patents are parsed by MATE. The resulting EPO corpus consists of roughly 4.6 billion tokens.

Task description. The task we address is semantic tagging of patents. The research reported here was conducted as part of a project on computational linguistics analysis of patent text. We want to be able to support functionalities like color-coding entities of a particular semantic class for quick perusal; or searching for entities in a particular semantic class. Our longterm goal is to support semantic tagging for a large variety of semantic classes. In this paper, we focus on the semantic classes SUBSTANCE and DISEASE. A substance is a particular kind of physical matter with uniform properties. Substances are of obvious relevance for the patent domain and a large proportion of patents contain substances. A disease is an abnormal condition that affects the body of an organism. We selected disease as a clearly nontechnical category to be able to investigate potential differences of lexical bootstrapping algorithms for categories with very different properties.

Gazetteers are crucial for good performance in machine-learning-based semantic tagging (Ratinov and Roth, 2009), e.g., the best performing systems for recognition of person, location and organization named entities all use gazetteer features (e.g., Florian et al. (2003)). It is in this context that we address the task of bootstrapping lexicons from corpora: for most semantic classes of interest in the patent domain high-coverage lexicons are not available.

Evaluation methodology. Since our primary task is semantic tagging, we evaluate the quality of the bootstrapped lexicon directly on this task, i.e., on the task of tagging members of the semantic class in text – rather than evaluating the lexicon in a type-based evaluation as a set of terms without context as most previous work has done. A tagging-based evaluation directly measures what we need for our application, e.g., frequent terms have a higher impact on tagging accuracy than rare terms, and ambiguous terms with a rare class sense depress tagging accuracy.

Terminology. From now on, we use the term *MWE* for a noun phrase that we identify as a candidate class instance; we include one-word noun phrases in the definition of MWE in this paper. We call an MWE in a particular context in our gold standard a *gold-standard MWE* if it was annotated as a member of the semantic class in question. We call an MWE a *lexicon MWE* if our bootstrapping algorithm has added it to the induced lexicon as a class instance.

Gold standard creation. Asking human annotators to mark all instances of SUBSTANCE/DISEASE in a randomly selected set of patents is very inefficient because this would result in annotators spending a lot of time reading patent text that contains (almost) no class instance. Moreover, annotation quality is higher in patents that contain at least a moderate number of class instances since annotators will remain alert as they go through the document.

To address this problem, for the two classes we stratify the EPO corpus into three strata according

[1] www.epo.org

to density ρ: high, medium and low. ρ is computed as the proportion of class instances per token. Since the low-density stratum contains virtually no class instances, we exclude it from our experiments.

We randomly select 1000 patents from each of the medium-density and high-density strata and then one sentence from each patent. One annotator labeled 200 sentences using the GATE[2] annotation tool. Then problematic annotation examples were discussed. Afterwards, the annotator labeled the remaining 1800 sentences. For assessing the quality of the gold standard, a second trained annotator labeled 200 sentences of our evaluation set. Inter-annotator agreement for both classes was $\kappa = .712$ (macro kappa) and $\kappa = .818$ (micro kappa) (Cohen, 1960), which indicates substantial to excellent agreement (Landis and Koch, 1977).

3 Bootstrapping algorithms

```
 1: lexicon $\leftarrow$ seed
 2: for int $i = 0$; $i < m$; $i$++ do
 3:     patterns $\leftarrow$patternsOf(lexicon)
 4:     score(patterns)
 5:     patterns $\leftarrow$ return-top-k(patterns,$20 + i$)
 6:     terms $\leftarrow$ termsOf(patterns) $-$ lexicon
 7:     score(terms)
 8:     lexicon $\leftarrow$ lexicon $\cup$ return-top-k(terms,5)
 9: end for
10: return  lexicon
```

Figure 1: Basilisk algorithm. The original version of Basilisk defines terms as head nouns.

The basic bootstrapping algorithm we use is Basilisk as shown in Figure 1 (Thelen and Riloff, 2002). Basilisk first initializes the lexicon as the seed set (line 1). The basic idea of the algorithm is to identify context patterns that reliably identify lexicon terms (lines 3–4), e.g., made of <X>. Line 5 selects a subset of patterns[3] based on the scoring function RlogF:

$$\text{RlogF}(\text{pattern}_i) = F_i/N_i \log_2(F_i)$$

where F_i is the number of learned lexicon terms that occur in pattern$_i$ and N_i is the total number of terms occurring in pattern$_i$. Lines 6–7 select the terms associated with the patterns selected on line 5 and score them. Terms are scored using AvgLog, the average log frequency, (line 7):

$$\text{AvgLog}(\text{term}_i) = 1/P_i \sum_{j=1}^{P_i} \log_2(F_j + 1)$$

where P_i is the number of patterns in which term$_i$ occurs and F_j is the number of learned lexicon terms that occur in pattern$_j$. The 5 highest scoring terms are then added to the lexicon (line 8).

3.1 Basilisk-G

To process the large-scale patent corpus efficiently, we implemented a simple chunker that identifies noun phrases (NPs) using regular expressions (REs) on POS tags. We refer to this RE/POS-based algorithm as *Basilisk-G* – for "Basilisk General Patterns". Basilisk-G is an instantiation of the basic Basilisk algorithm in Figure 1 that uses a more general definition of patterns that only requires POS tagging and no parsing. We use all patterns of the form $w_{-i} \ldots w_{-1}$<np>$w_1 \ldots w_j$ where $0 \leq i, j \leq 3$ and $i + j \geq 2$; i.e., context extends up to three tokens out to the left and right from <np> and must consist of at least two tokens. We discard patterns whose context does not contain a verb or a noun. The standard version of Basilisk uses pattern templates like <subj> verb and noun prep <np>. Our more general definition covers most instantiations of these templates, but it extends the patterns considered to a much larger variety of lexical contexts. For example, fragments of a coordination like , silver, <np> or platinum are also instantiations of the general Basilisk-G pattern template. As we will see later, these types of patterns (which original Basilisk does not use) turn out to be very effective.

Similar to patterns, we define MWEs in Basilisk-G (Figure 1, line 6) as part of an NP extracted using REs: an MWE is a (possibly zero-length) sequence of prehead modifiers (adjectives and nouns) terminated by the head.

3.2 Basilisk-C

In this section we introduce *Basilisk-C* – for "Basilisk Coordination Patterns" , a Basilisk instantiation that uses only coordinations.

We allow two different types of coordinations: *and/or* coordinations and punctuation coordinations.

An *and/or* coordination is a list of NPs consisting of two parts. In the first part of the list, NPs are separated by commas or semicolons. In the second part, NPs are separated by "and", "or" or "and/or". The second part has minimum length 2:

((NP,)*|(NP;)*) NP ((and(/or)?|or) NP)+

[2] gate.ac.uk

[3] We discard patterns that only occur with already learned terms, guaranteeing that each of the selected $20 + i$ patterns on line 5 can potentially contribute new terms

A punctuation coordination is defined as a list of at least three NPs separated by commas or semicolons: ((NP,)+ NP, NP) | ((NP;)+ NP; NP)[4]

Because we detect the coordinations and the NPs based on POS REs without performing a full syntactic analysis and because the assumption of co-hyponymy is incorrect in certain cases, there are several incorrect matches. We automatically removed border elements of a coordination if they indicate an extraneousness to the coordination. One indicator of extraneousness was the unique presence of a determiner for example "this description" in "as concrete examples of [this description, methyl alcohol and benzyl alcohol] may be cited", where [. . .] matches the coordination pattern. Moreover, we removed conjuncts that indicate a hypernym relation such as "other products" in "copolymers, polyisobutene and other products".

We treat the conjuncts that survive filtering as an unordered set, i.e., we ignore their order in the text. The set is discarded and not used by Basilisk-C if it only contains one element.

4 Experimental setup and results

Experimental setup. We evaluate performance of the two algorithms Basilisk-G and Basilisk-C introduced above. We run experiments on EPO (Section 2) with the goal of learning the classes SUBSTANCE and DISEASE. Our seed set (Figure 1, line 1) consists of the 4223 substances distributed by Ciaramita and Johnson (2003) as part of SuperSenseTagger and 239 diseases extracted from Simple English Wikipedia[5].

For Basilisk-C, we extracted 9.7 million unique coordinations, out of a total of 25 million.

For Basilisk-G, we found 1.6 billion unique context patterns. In order to be able to run experiments quickly, we introduce frequency thresholds for MWEs, patterns and MWE-pattern combinations. We only consider MWEs and patterns that occur at least $\theta_1 = 10$ times and MWE-pattern combinations that occur at least $\theta_2 = 3$ times in EPO. These thresholds are unlikely to diminish lexicon quality since many rare instances of MWEs are due to OCR errors or failures of our RE-based recognition of NPs (see also (Qadir and Riloff, 2012)). Using the thresholds θ_1 and θ_2, there were 3.2 million unique MWEs, 56 million unique patterns and 121 million unique MWE-pattern combinations. This is the raw data we run Basilisk-G on.

As discussed in Section 2, our evaluation methodology directly evaluates the semantic lexicon on the task of interest: semantic tagging of patents. The tagging method we use is simple lexicon lookup. While tagging MWEs we exploit the compositional structure of entities by merging adjacent or overlapping token-based labels (e.g., *fatty acid* and *acid amide* are merged to *fatty acid amide*). In our decision to use lexicon lookup for tagging, we follow Qadir and Riloff (2012), who argue convincingly that for a specialized class and domain, ambiguity of terms (which would be the main reason for using a context-dependent method like a CRF) is a limited phenomenon and ignoring it does not greatly affect performance. Even so, it is important to keep in mind that tagging precision does not directly reflect lexicon accuracy.

We use the measures precision, recall and F_1. Tagging results are evaluated using the evaluation module of GATE. The scores give half credits for partial matches and full credits for exact matches.

Performance of Basilisk-G and Basilisk-C. Table 1 shows the performance of the baseline and of Basilisk-G and Basilisk-C for different lexicon sizes. The baseline uses the seed set (SUBSTANCE: 4223; DISEASE: 239) for tagging. We first run iterations until the size of the induced lexicon is 5000 and then double the lexicon three times – to 10,000, 20,000 and 40,000 – to investigate the relationship between lexicon size and tagging performance.

Both Basilisk-G and Basilisk-C consistently beat the baseline in recall and F_1 for DISEASE and in all three measures for SUBSTANCE. We mark each performance number with a star if it is significantly higher than the number above it.[6] For example, Basilisk-G's and Basilisk-C's F_1 of .549 for 10,000 substances is significantly better than the baseline (.539).

Basilisk-C outperforms Basilisk-G in most cases. We mark each Basilisk-C performance number with † if it is significantly higher than the Basilisk-G number to the left of it. The largest differences between Basilisk-C and Basilisk-G can be found for the smaller semantic class of diseases and at larger lexicon sizes. This is to be expected

[4] This version of Basilisk uses the same RE to detect NPs as Basilisk-G.

[5] simple.wikipedia.org/wiki/List_of_diseases

[6] Approximate randomization test (Yeh, 2000), $p < .05$

size	SUBSTANCE						DISEASE					
	P		R		F_1		P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539		.793		. 233		.360	
5000	.599*	.598	.494*	.492*	.542*	.540*	.790	.724	.455*	.556*†	.578*	.629*†
10,000	.605*	.604*	.502*	.504*	.549*	.549*	.476	.643†	.645*	.602*	.548	.622†
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†	.392	.530†	.642	.701*†	.487	.604†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†	.300	.473†	.642	.720†	.409	.571†

Table 1: Tagging performance measured by precision (P), recall (R) and F_1 of seed baseline and for different lexicon sizes of Basilisk-G (B-G) and Basilisk-C (B-C); * indicates significantly higher than the number above it; † indicates significantly higher than the number to the left of it.

because we would expect semantic drift to be more prominent for smaller classes and to grow with the size of the induced lexicon. Closer inspection reveals that Basilisk-G indeed drifts to any kind of technical properties. For substances, Basilisk-C outperforms Basilisk-G mainly in recall. This large class is less sensitive to semantic drift but, as we will discuss in detail in the error analysis, still benefits from the MWE extraction of Basilisk-C. These results support the argument we have made for restricting to coordinations in Basilisk for both predominant classes such as SUBSTANCE and minor semantic classes such as DISEASE.

Note that the best performance for the small class of diseases is found at a lexicon size of 5000: $F_1 = .629$. It outperforms the seed baseline (+.269) and Basilisk-G (+.051). Precision drops rapidly when doubling the lexicon size and introducing more and more semantic drift. For the large semantic class of substances we find that increasing the lexicon size generally improves performance. The overall best result achieved, $F_1 = .582$, is achieved by Basilisk-C and the largest lexicon size (40,000).

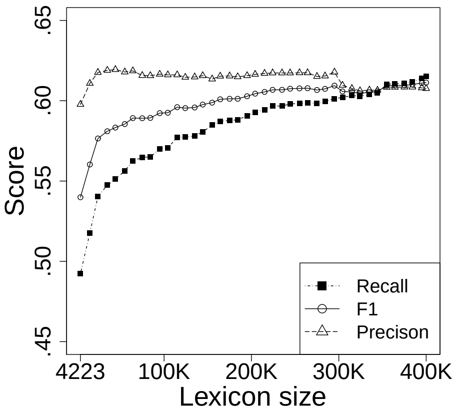

Figure 2: Performance of Basilisk-C as a function of lexicon size for substances

Our comparisons between Basilisk-G and Basilisk-C are limited to a lexicon size of 40,000 because running Basilisk-G for larger lexicons becomes infeasible in our current setup. However, one of the additional advantages of restricting to coordinations is scalability. Figure 2 shows the performance of Basilisk-C as a function of lexicon size for very large substance lexicons. The figure suggests that there is an upward trend for F_1 and recall up to very large lexicon sizes. The curves do not increase monotonically, partly due to the fact that once the lexicon has reached a size of more than 100,000, only a few additional MWEs found in the evaluation corpus are responsible for the changes. Thus, the curves do not directly reflect actual expected performance. Basilisk-C achieves the highest performance at the right edge of the graph at lexicon size 400,000: $F_1 = .611$, P$= .608$, R$= .615$. To our knowledge, Basilisk-type algorithms have not been run for lexicons of this size before.

The tagging performance seems low for practical purposes. However, this reflects the lexicon quality (i.e., the bootstrapping performance) only partially. Since we are using a large domain-specific patent corpus, we are faced with many high-specific and infrequent terms. Many false positives arise because of class instances missed by the annotators. An optimal gold standard for this domain would require domain specialists.

Comparison with original Basilisk. We already explained that running original Basilisk on the whole EPO corpus is not feasible due to its size and the length of the sentences. However, for a comparison of Basilisk-G and Basilisk-C with the original Basilisk, we parsed sentences up to a size of 100 tokens from 25,000 sample patents. We define *Basilisk-LS* – for "Basilisk Lexico-Syntactic patterns"and extract all lexico-syntactic patterns described in (Riloff and Phillips, 2004). We introduce two versions of Basilisk-LS: the originally head-based Basilisk-LS$_{head}$ and a variant, Basilisk-LS$_{MWE}$, that extracts MWEs defined as basic noun phrases without determiner or postnominal phrases such as PPs or relative clauses. From all parsed sentences, we extract POS-based coordinations and all general patterns for our systems. The tagging results for a substance lexicon of size 20K is shown in Table 2.

System	P	R	F_1
Basilisk-LS$_{head}$	.437	.502	.467
Basilisk-LS$_{MWE}$	.582	.506	.541
Basilisk-G	.560	.524	.542
Basilisk-C	.620	.567	.592

Table 2: Comparison of B-LS, B-G and B-C

The results for Basilisk-LS$_{head}$ show that precision is a severe problem in tagging substance head nouns on patents. When inducing and tagging MWEs as it has been done by the other systems, precision is much higher. Both Basilisk-LS$_{MWE}$ and Basilisk-G only slightly outperform the seed baseline in F_1 (Table 1). Once more Basilisk-C outperforms all other systems by far. [7]

Extension on other classes and domains. To show that Basilisk-C can be applied to other classes and domains, we ran an additional experiment. We applied Basilisk-C to the class FIXED-LOCATION as defined by Qadir and Riloff (2012) and used Wikipedia coordinations. As there was no annotated data available for this class, we ran a type-based evaluation. A human judge rates accuracy of 100 samples of the lexicon. As seed set, we selected five US states, five European countries and ten national capitals. We induced 5000 fixed locations with an accuracy of 80%. We conclude that depending on corpus size lexical bootstrapping based on coordinations is applicable to any domain for several classes.

5 Analysis and discussion

High arity. In previous sections we explained that the high arity of the coordination relation constrains the semantics of its arguments and remedies problems related to ambiguity that can give rise to semantic drift. We have seen this in the results. The relatively small class (for our domain) of diseases is prone to semantic drift especially when larger lexicon sizes are induced. Basilisk-C is able to remedy this problem and leads to higher performances than Basilisk-G and the differences are largest for larger lexicon sizes.

On the other hand, we discussed the phenomenon that MWEs in short coordinations tend to be less specific. Despite the fact that shorter coordinations provide a smaller pool of term candidates, we expect recall in the semantic tagging task to be higher when using shorter coordinations because the available term candidates are less specific and thus have a higher frequency, i.e., a higher chance to occur in the test set. Table 3 shows the tagging performance of a lexicon with 20,000 substances and one with 10,000 diseases induced by Basilisk-C applied to different ranges of coordination lengths. It shows that Basilisk-C using only coordinations up to a size of 5 terms ("2 to 5") outperforms Basilisk-C using all coordinations ("2 to ∞") in recall. In predominant classes such as SUBSTANCE, shorter coordinations do not harm precision. However, for classes like DISEASE, precision decreases when shorter coordinations are used, as illustrated in Table 3.

Coordination length	P	R	F_1
20,000 substances			
2 to ∞	.614	.529	.568
2 to 5	.615	.568	.591
10,000 diseases			
2 to ∞	.643	.602	.622
2 to 5	.470	.645	.544

Table 3: Comparison of coordination lengths

High-confidence pattern. We argued in subsection 3.2 that coordinations impose a stronger semantic coherence on MWEs than general context patterns or lexico-syntactic patterns do. Table 4 shows that coordinations are indeed high-confidence patterns for learning substances. High-confidence Basilisk-G patterns after 1000 and 6000 iterations are listed. Each of the top 20 patterns after 1000 iterations is a coordination. Apparently, the patterns that are selected in the beginning of learning as the ones best suited for identifying substances are all fragments of coordinations. Thus, performance of Basilisk-G and Basilisk-C for a lexicon of 10,000 MWEs (cf. Table 1) is fairly equal.

In contrast, after 6000 iterations only three of the highest-confidence patterns still are coordinations (not shown) – the other 17 are other types of

[7] Recall of Basilisk-C for the subcorpus is even better than on the full EPO corpus shown in Table 1. The reason for this is that sentences up to 100 tokens contain shorter coordinations. MWEs in short coordinations tend to be less specific and thus more frequent in a test set. This explains why recall of semantic tagging is better when using shorter coordinations.

	type of pattern	example
$i=1000$	NN , <np> ,	nitrate , <np> ,
	, <np> , NN	, <np> , magnesium
	NN , <np> CC	oxide , <np> , and
$i=6000$	, <np> , NN	, <np> , sodium
	NN of <np> (	weight of <np> (
	of <np> verb	of <np> was added

Table 4: Highest-confidence Basilisk-G patterns after i iterations (examples from top 20)

patterns (Table 4, $i = 6000$). As Basilisk-G's performance improves more slowly than performance of Basilisk-C after the initial iterations (cf. Table 1, 20,000 to 40,000), we conclude that coordinations are the most effective patterns and the addition of other pattern types contribute little to learning new substances.

Scalability. Besides the scalability upgrade in preprocessing by avoiding parsing, Basilisk-C, in particular ranking of coordinations and MWEs, runs quicker as the input of term-pattern combinations is about 80% smaller than for original Basilisk. This means a crucial scalability benefit for corpora even larger than EPO.

MWE extraction. The results in Table 2 show that inducing and tagging only head nouns rather than MWEs leads to poor precision. This result is to be expected as our set of gold-standard MWEs comprises 45.4% true MWEs and tagging only the head nouns thereof leads to partial credits.

By restricting patterns to coordinations, Basilisk-C avoids MWE recognition problems. Coordinated noun phrases tend to be less modified, less complex and the context of an NP within the coordination makes it easier to determine its boundaries; the internal boundaries are always connectors. Table 5 shows some MWEs induced by Basilisk-G and the subparts thereof induced by Basilisk-C.

high-molecular weight vinylidene fluoride resin
above metal chelate compound
unsaturated fatty acid ester
heat-fusible polymer fine particle

Table 5: Examples of MWEs in B-G; underlined tokens match MWEs induced by B-C

Coordination abundance. Basilisk-C works best for a text type in which large coordinations are abundant since this is the only context pattern it considers. In an analysis of the prevalence of coordinations in different corpora, we observed that long coordinations (those with at least three conjuncts) are more prevalent in patents than in other genres (average length of 4.6 in EPO vs 3.6 in other corpora). Thus, coordinations seem to be a particularly promising resource for lexical bootstrapping in technical domains like patents. However, as exemplified by our experiments with Wikipedia, Basilisk-C shows similar performance on other domains, given that the members of the semantic class appear often in coordinations.

6 Related work

We have chosen a semisupervised approach to lexical bootstrapping here since it is reasonable to expect that in the type of application scenario we have in mind resources are available to create a seed set. There are also completely unsupervised approaches to lexical bootstrapping (e.g., Lin and Pantel (2002); Davidov et al. (2007); Van Durme and Paşca (2008); Dalvi et al. (2012)), but they usually cannot match the quality of approaches like ours that use human input such as a seed set.

The bootstrapping approach we have adopted here starts with a seed set and then iteratively extends the lexicon by adding the highest-confidence MWEs in each iteration. Basilisk (Thelen and Riloff, 2002) is perhaps the best known bootstrapping method of this type, but there exists a large literature on similar methods, some of which exploit lexical co-occurrence statistics (e.g., Riloff and Shepherd (1997)) and some of which use syntactic analysis (e.g., Roark and Charniak (1998); Riloff and Jones (1999); Phillips and Riloff (2002)). Our approach does not make use of syntactic analysis but relies on POS patterns.

Some recent work attempts to improve Basilisk's accuracy. Igo and Riloff (2009) enhance precision by checking candidate terms using web queries. Qadir and Riloff (2012) combine Basilisk in an ensemble with an SVM tagger and a coreference resolution system. Our focus is learning technical terminology from very large corpora using coordinations, but any work that improves the accuracy of basic Basilisk could also be beneficial in our setting.

Gazetteers are crucial for good performance in named entity recognition (NER). Work on automatic extraction of gazetteers for NER includes (Toral and Muñoz, 2006; Kazama and Torisawa, 2007). Most of this work is complementary to our approach because it uses knowledge bases like Wikipedia or is only applicable to traditional named entities (NEs). Traditional NEs like person are capitalized. Substances are not. Our work also differs in its focus on coordinations and technical text.

Coordinations have been frequently used in work on lexical acquisition. Caraballo (1999) builds a hierarchy of coordinated nouns and their hypernyms. Cederberg and Widdows (2003) use coordinations to estimate the semantic relatedness of nouns. Widdows and Dorow (2002) and Qiu et al. (2011) cluster nouns and evaluate the semantic homogeneity of the clusters. Etzioni et al. (2005) use Hearst patterns to bootstrap lexicons. They also consider coordinations when selecting candidates. This previous work on coordinations is unsupervised and not focused on learning a particular semantic class that is defined by a seed set.

Roark and Charniak (1998) use a variety of syntactic constructions, including coordinations, for bootstrapping. Our approach is different in that we do not require parsing and that we cover MWEs in general, not just heads or compounds with a common head. However, some of the other syntactic constructions presented by Roark and Charniak (1998) could also be amenable to reliable detection by REs. We plan to investigate this in future work. Goyal et al. (2010) create a plot unit representation creator. Therefore, they induce a lexicon of *patient polarity verbs* (i.e., verbs that impart positive or negative states on their patients) based on Basilisk, that learns from coordinated verbs. This work is focused on verbs with the same patient polarity in binary coordinations extracted from a web corpus. Our approach is based on coordinations of any size from a large patent corpus and focuses on semantic lexicon induction.

One distinguishing characteristic of our work is the patent domain. Other work on technical or scientific domains includes press releases of pharmaceutical companies (Phillips and Riloff, 2002), medline abstracts (McIntosh and Curran, 2009), message board posts from the Veterinary Information Network (Huang and Riloff, 2010) and texts from ProMed and PubMed (Igo and Riloff, 2009; Qadir and Riloff, 2012).

Patents can be argued to be particularly difficult technical text due to long sentences, legalese and complex NP syntax. To the best of our knowledge, our experiments are also the largest semantic bootstrapping experiments on technical text to date. While there has been much work on experiments on large web corpora and other general text (e.g., Kozareva et al. (2008); Carlson et al. (2009); Bakalov et al. (2011)), the corpora in other lexical bootstrapping work on technical domains have been an order of magnitude smaller than ours.

We showed that using only coordinations remedies the problem of semantic drift. Other work on semantic drift includes Yangarber et al. (2002); Curran et al. (2007); McIntosh and Curran (2008); McIntosh and Curran (2009).

7 Conclusion

In this paper, we presented Basilisk-C. The method is inspired by original Basilisk but adapts it to large corpora of technical text by restricting it to one type of patterns: coordinations.

This restriction to coordinations, a relation that is known to impose strong semantic coherence upon its members and as such a possible remedy for semantic drift, leads to significant improvements for the task of semantic tagging, compared to an unrestricted version of Basilisk.

We further extended original Basilisk to include MWEs, as these are predominant in technical text and showed that coordination patterns yield higher precision in MWE extraction.

The proposed method avoids the need for parsing, which is cumbersome for large corpora with long sentences, typical for the technical domain. In general, we upgrade scalibility because the coordination patterns represent a fraction of the patterns original Basilisk utilized.

Apart from using linguistics patterns such as coordinations, we plan to use structured data, such as table columns and rows to extract co-hyponyms in future work.

We will make our gold-standard and the induced lexicons publicly available[8].

Acknowledgments

This research was supported by the project TOPAS[9] (Tool platform for intelligent Patent Analysis and Summarization), its team members and the European Commission with its FP7-SME Program. The TOPAS Consortium is composed of five partners: Brügmann Software Inc., IALE Inc., Intellisemantic Inc., Pompeu Fabra University and University of Stuttgart. We thank all colleagues for their support. This research was further supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) as part of the SFB 732.

[8] www.ims.uni-stuttgart.de/data/basiliskc.resources.tgz

[9] topasproject.eu

References

A. Bakalov, A. Fuxman, P. P. Talukdar, and S. Chakrabarti. 2011. Scad: Collective discovery of attribute values. In *WWW 2011*.

B. Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING 2010*.

S. A. Caraballo. 1999. Automatic acquisition of a hypernym-labeled noun hierarchy from text. In *ACL 99*.

A. Carlson, J. Betteridge, E. R. Hruschka, and T. M. Mitchell. 2009. Coupling semi-supervised learning of categories and relations. In *NAACL-HTL 2009*.

S. Cederberg and D. Widdows. 2003. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *CoNLL 2002*.

M. Ciaramita and M. Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *EMNLP 2003*.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20:37–46.

J. R. Curran, T. Murphy, and B. Scholz. 2007. Minimising Semantic Drift with Mutual Exclusion Bootstrapping. In *PACLING 2007*.

B. Dalvi, W. W. Cohen, and J. Callan. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *WSDM 2012*.

D. Davidov, A. Rappoport, and M. Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *ACL 2007*.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165:91–134.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named entity recognition through classifier combination. In *HLT-NAACL 2003*.

A. Goyal, E. Riloff, and H. Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *EMNLP 2010*.

R. Huang and E. Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *ACL 2010*.

S. P. Igo and E. Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *NAACL 2009*, pages 18–26.

J. Kazama and K. Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL 2007*.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL 2008*.

J. R. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

D. Lin and P. Pantel. 2002. Concept discovery from text. In *COLING 2002*.

T. McIntosh and J. R. Curran. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *ALTA 2008*.

T. McIntosh and J. R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *ACL-IJCNLP 2009*.

W. Phillips and E. Riloff. 2002. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In *EMNLP 2002*.

A. Qadir and E. Riloff. 2012. Ensemble-based semantic lexicon induction for semantic tagging. In **SEM-2012*.

L. Qiu, Y. Wu, J. Shi, Y. Shao, and Z. Long. 2011. Induction of semantic classes based on coordinate patterns. In *WI-IAT 2011*.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL 2009*.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI-99*.

E. Riloff and W. Phillips. 2004. An introduction to the sundance and autoslog systems. Technical report.

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *EMNLP 1997*.

B. Roark and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *ACL 98*.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, September.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP 2002*.

A. Toral and R. Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *EACL 2006*.

B. Van Durme and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *AAAI 2008*.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *COLING 2002*.

R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised learning of generalized names. In *COLING 2002*.

A. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000*.