GlossBoot: Bootstrapping Multilingual Domain Glossaries from the Web

Flavio De Benedictis, Stefano Faralli and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
flavio.debene@gmail.com,{faralli,navigli}@di.uniroma1.it

Abstract

We present GlossBoot, an effective minimally-supervised approach to acquiring wide-coverage domain glossaries for many languages. For each language of interest, given a small number of hypernymy relation seeds concerning a target domain, we bootstrap a glossary from the Web for that domain by means of iteratively acquired term/gloss extraction patterns. Our experiments show high performance in the acquisition of domain terminologies and glossaries for three different languages.

1 Introduction

Much textual content, such as that available on the Web, contains a great deal of information focused on specific areas of knowledge. However, it is not infrequent that, when reading a domain-specific text, we humans do not know the meaning of one or more terms. To help the human understanding of specialized texts, repositories of textual definitions for technical terms, called glossaries, are compiled as reference resources within each domain of interest. Interestingly, electronic glossaries have been shown to be key resources not only for humans, but also in Natural Language Processing (NLP) tasks such as Question Answering (Cui et al., 2007), Word Sense Disambiguation (Duan and Yates, 2010; Faralli and Navigli, 2012) and ontology learning (Navigli et al., 2011; Velardi et al., 2013).

Today large numbers of glossaries are available on the Web. However most such glossaries are small-scale, being made up of just some hundreds of definitions. Consequently, individual glossaries typically provide a partial view of a given domain. Moreover, there is no easy way of retrieving the subset of Web glossaries which appertains to a domain of interest. Although online services such as Google Define allow the user to retrieve definitions for an input term, such definitions are extracted from Web glossaries and put together for the given term regardless of their domain. As a result, gathering a large-scale, full-fledged domain glossary is not a speedy operation.

Collaborative efforts are currently producing large-scale encyclopedias, such as Wikipedia, which are proving very useful in NLP (Hovy et al., 2013). Interestingly, wikipedias also include manually compiled glossaries. However, such glossaries still suffer from the same above-mentioned problems, i.e., being incomplete or over-specific,[1] and hard to customize according to a user's needs.

To automatically obtain large domain glossaries, over recent years computational approaches have been developed which extract textual definitions from corpora (Navigli and Velardi, 2010; Reiplinger et al., 2012) or the Web (Fujii and Ishikawa, 2000). The former methods start from a given set of terms (possibly automatically extracted from a domain corpus) and then harvest textual definitions for these terms from the input corpus using a supervised system. Web-based methods, instead, extract text snippets from Web pages which match pre-defined lexical patterns, such as "X is a Y", along the lines of Hearst (1992). These approaches typically perform with high precision and low recall, because they fall short of detecting the high variability of the syntactic structure of textual definitions. To address the low-recall issue, recurring cue terms occurring within dictionary and encyclopedic resources can be automatically extracted and incorporated into lexical patterns (Saggion, 2004). However, this approach is term-specific and does not scale to arbitrary terminologies and domains.

In this paper we propose GlossBoot, a novel approach which reduces human intervention to a bare minimum and exploits the Web to learn a

[1] http://en.wikipedia.org/wiki/Portal:Contents/Glossaries

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 528–538,
Sofia, Bulgaria, August 4-9 2013.

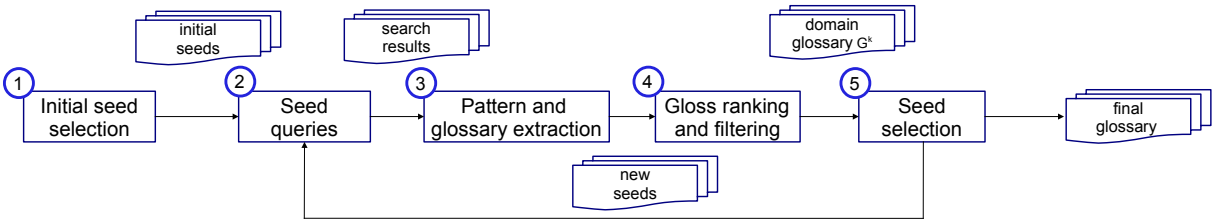

Figure 1: The GlossBoot bootstrapping process for glossary learning.

full-fledged domain glossary. Given a domain and a language of interest, we bootstrap the glossary learning process with just a few hypernymy relations (such as computer *is-a* device), with the only condition that the (term, hypernym) pairs must be specific enough to implicitly identify the domain in the target language. Hence we drop the requirement of a large domain corpus, and also avoid the use of training data or a manually defined set of lexical patterns. To the best of our knowledge, this is the first approach which jointly acquires large amounts of terms and glosses from the Web with minimal supervision for any target domain and language.

2 GlossBoot

Our objective is to harvest a domain glossary G containing pairs of terms/glosses in a given language. To this end, we automatically populate a set of HTML patterns P which we use to extract definitions from Web glossaries. Initially, both $P := \emptyset$ and $G := \emptyset$. We incrementally populate the two sets by means of an initial seed selection step and four iterative steps (cf. Figure 1):

Step 1. Initial seed selection: first, we manually select a set of K hypernymy relation seeds $S = \{(t_1, h_1), \ldots, (t_K, h_K)\}$, where the pair ($t_i$, h_i) contains a term t_i and its generalization h_i (e.g., (*firewall*, *security system*)). This is the only human input to the entire glossary learning process. The selection of the input seeds plays a key role in the bootstrapping process, in that the pattern and gloss extraction process will be driven by these seeds. The chosen hypernymy relations thus have to be as topical and representative as possible for the domain of interest (e.g., (*compiler*, *computer program*) is an appropriate pair for computer science, while (*byte*, *unit of measurement*) is not, as it might cause the extraction of several glossaries of units and measures).

We now set the iteration counter k to 1 and start the first iteration of the glossary bootstrapping process (steps 2-5). After each iteration k, we keep track of the set of glosses G^k, acquired during iteration k.

Step 2. Seed queries: for each seed pair (t_i, h_i), we submit the following query to a Web search engine: "t_i" "h_i" `glossaryKeyword`[2] (where `glossaryKeyword` is the term in the target language referring to *glossary* (i.e., *glossary* for English, *glossaire* for French etc.)) and collect the top-ranking results for each query.[3] Each resulting page is a candidate glossary for the domain implicitly identified by our relation seeds S.

Step 3. Pattern and glossary extraction: we initialize the glossary for iteration k as follows: $G^k := \emptyset$. Next, from each resulting page, we harvest all the text snippets s starting with t_i and ending with h_i (e.g., "*firewall* – a <i>*security system*" where t_i = *firewall* and h_i = *security system*), i.e., $s = t_i \ldots h_i$. For each such text snippet s, we perform the following substeps:

(a) extraction of the term/gloss separator: we start from t_i and move right until we extract the longest sequence p_M of HTML tags and non-alphanumeric characters, which we call the *term/gloss separator*, between t_i and the glossary definition (e.g., " –" between "firewall" and "a" in the above example).

(b) gloss extraction: we expand the snippet s to the right of h_i in search of the entire gloss of t_i, i.e., until we reach a block element (e.g., , <p>,), while ignoring formatting elements such as , <i> and <a> which are typically included within a definition sentence. As a result, we obtain the sequence $t_i\ p_M\ gloss_s(t_i)\ p_R$, where $gloss_s(t_i)$ is our gloss for seed term t_i in snippet s (which includes h_i by construction) and p_R is the HTML block element

[2] In what follows we use the `typewriter font` for keywords and term/gloss separators.

[3] We use the Google Ajax APIs, which return the 64 top-ranking search results.

Generalized pattern	HTML text snippet
<**strong**> * </**strong**> **-** * </**span**>	<**strong**>Interrupt</**strong**> **-** The suspension of normal program execution to perform a higher priority service routine as requested by a peripheral device. </**span**>
<**dt**> * </**dt**><**dd**> * </**dd**>	<**dt**>Netiquette</**dt**><**dd**>The established conventions of online politeness are called netiquette.</**dd**>
<**h3**> * </**h3**><**p**> * </**p**>	<**h3**>Compiler</**h3**><**p**>A program that translates source code, such as C++ or Pascal, into directly executable machine code.</**p**>
<**span**> * </**span**> **-** * </**p**>	<**span**>Signature</**span**> **-** A function's name and parameter list. </**p**>
<**span**> * </**span**>**:** * <**span**>	<**span**>Blog</**span**>**:** Short for "web log", a blog is an online journal. <**span**>

Table 1: Examples of generalized patterns together with matching HTML text snippets.

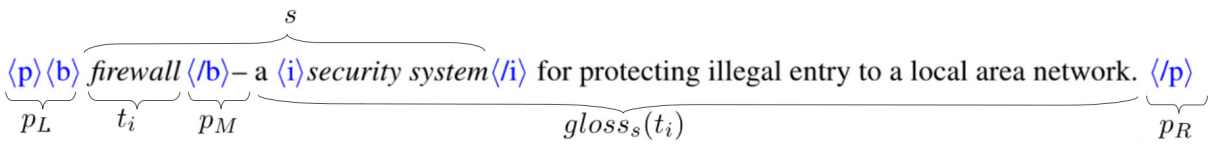

Figure 2: An example of decomposition during pattern extraction for a snippet matching the seed pair (*firewall*, *security system*).

to the right of the extracted gloss. In Figure 2 we show the decomposition of our example snippet matching the seed (*firewall*, *security system*).

(c) pattern instance extraction: we extract the following pattern instance:

$$p_L\ t_i\ p_M\ gloss_s(t_i)\ p_R,$$

where p_L is the longest sequence of HTML tags and non-alphanumeric characters obtained when moving to the left of t_i (see Figure 2).

(d) pattern extraction: we generalize the above pattern instance to the following pattern:

$$p_L\ *\ p_M\ *\ p_R,$$

i.e., we replace t_i and $gloss_s(t_i)$ with *. For the above example, we obtain the following pattern:

 * – * .

Finally, we add the generalized pattern to the set of patterns P, i.e., $P := P \cup \{p_L\ *\ p_M\ *\ p_R\}$. We also add the first sentence of the retrieved gloss $gloss_s(t_i)$ to our glossary G^k, i.e., $G^k := G^k \cup \{(t_i, first(gloss_s(t_i)))\}$, where $first(g)$ returns the first sentence of gloss g.

(e) pattern matching: finally, we look for additional pairs of terms/glosses in the Web page containing the snippet s by matching the page against the generalized pattern $p_L\ *\ p_M\ *\ p_R$. We then add to G^k the new (term, gloss) pairs matching the generalized pattern. In Table 1 we show some non-trivial generalized patterns together with matching HTML text snippets.

As a result of step 3, we obtain a glossary G^k for the terms discovered at iteration k.

Step 4. Gloss ranking and filtering: importantly, not all the extracted definitions pertain to the domain of interest. In order to rank the glosses obtained at iteration k by domain pertinence, we assume that the terms acquired at previous iterations belong to the target domain, i.e., they are domain terms at iteration k. Formally, we define the terminology T_1^{k-1} of the domain terms accumulated up until iteration $k-1$ as follows: $T_1^{k-1} := \bigcup_{i=1}^{k-1} T^i$, where $T^i := \{t : \exists(t,g) \in G^i\}$. For the base step $k = 1$, we define $T_1^0 := \{t : \exists(t,g) \in G^1\}$, i.e., we use the first-iteration terminology itself.

To rank the glosses, we first transform each acquired gloss g to its bag-of-word representation $Bag(g)$, which contains all the single- and multi-word expressions in g. We use the lexicon of the target language's Wikipedia together with T_1^{k-1} in order to obtain the bag of content words.[4] Then we

[4] In fact Wikipedia is only utilized in the multi-word identification phase. We do not use Wikipedia for discovering new terms.

Term	Gloss	Hypernym	# Seeds	Score
dynamic packet filter	A **firewall** *facility* that can monitor the *state* of active connections and use this information to determine which network packets to allow through the **firewall**	firewall	2	0.75
die	An integrated *circuit* **chip** cut from a finished wafer.	integrated circuit	1	0.75
constructor	a method used to help create a new object and initialise its data	method	0	1.00

Table 2: Examples of extracted terms, glosses and hypernyms (seeds are in bold, domain terms, i.e., in T_1^{k-1}, are underlined, non-domain terms in italics).

calculate the domain score of a gloss g as follows:

$$score(g) = \frac{|Bag(g) \cap T_1^{k-1}|}{|Bag(g)|}. \quad (1)$$

Finally, we use a threshold θ (whose tuning is described in the experimental section) to remove from G^k those glosses g whose $score(g) < \theta$.

In Table 2 we show some glosses in the computer science domain (second column, domain terms are underlined) together with their scores (last column).

Step 5. Seed selection for next iteration: we now aim at selecting the new set of hypernymy relation seeds to be used to start the next iteration. We perform three substeps:

(a) Hypernym extraction: for each newly-acquired term/gloss pair $(t, g) \in G^k$, we automatically extract a candidate hypernym h from the textual gloss g. To do this we use a simple unsupervised heuristic which just selects the first term in the gloss.[5] We show an example of hypernym extraction for some terms in Table 2 (we report the term in column 1, the gloss in column 2 and the hypernyms extracted by the first term hypernym extraction heuristic in column 3).

(b) (Term, Hypernym)-ranking: we sort all the glosses in G^k by the number of seed terms found in each gloss. In the case of ties (i.e., glosses with the same number of seed terms), we further sort the glosses by the score given in Formula 1. We show an example of rank for some glosses in Table 2, where seed terms are in bold, domain terms (i.e., in T_1^{k-1}) are underlined, and non-domain terms are shown in italics.

[5]While more complex strategies could be used, such as supervised classifiers (Navigli and Velardi, 2010), we found that this heuristic works well because, even when it is not a hypernym, the first term plays the role of a cue word for the defined term.

(c) New seed selection: we select the (term, hypernym) pairs corresponding to the K top-ranking glosses.

Finally, if k equals the maximum number of iterations, we stop. Else, we increment the iteration counter (i.e., $k := k + 1$) and jump to step (2) of our glossary bootstrapping algorithm after replacing S with the new set of seeds.

The output of glossary bootstrapping is a domain glossary $G := \bigcup_{i=1,...,max} G^i$, which includes a domain terminology $T := \{t : \exists(t, g) \in G\}$ (i.e., $T := T_1^{max}$) and a set of glosses $glosses(t)$ for each term $t \in T$ (i.e., $glosses(t) := \{g : \exists(t, g) \in G\}$).

3 Experimental Setup

3.1 Domains and Gold Standards

For our experiments we focused on four different domains, namely, Computing, Botany, Environment, and Finance, and on three languages, namely, English, French and Italian. Note that not all the four domains are clear-cut. For instance, the Environment domain is quite interdisciplinary, including terms from fields such as Chemistry, Biology, Law, Politics, etc.

For each domain and language we selected as gold standards well-reputed glossaries on the Web, such as: the Utah computing glossary,[6] the Wikipedia glossary of botanical terms,[7] a set of Wikipedia glossaries about environment,[8] and the Reuters glossary for Finance[9] (full list at `http://lcl.uniroma1.it/glossboot/`). We report the size of the four gold-standard datasets in Table 4.

[6]http://www.math.utah.edu/∼wisnia/glossary.html

[7]http://en.wikipedia.org/wiki/Glossary_of_botanical_terms

[8]http://en.wikipedia.org/wiki/List_of_environmental_issues, http://en.wikipedia.org/wiki/Glossary_of_environmental_science, http://en.wikipedia.org/wiki/Glossary_of_climate_change

[9]http://glossary.reuters.com/index.php/Main_Page

Computing		Botany		Environment		Finance	
chip	circuit	leaf	organ	sewage	waste	eurobond	bond
destructor	method	grass	plant	acid rain	rain	asset play	stock
compiler	program	cultivar	variety	ecosystem	system	income stock	security
scanner	device	gymnosperm	plant	air monitoring	sampling	financial intermediary	institution
firewall	security system	flower	reproductive organ	global warming	temperature	derivative	financial product

Table 3: Hypernymy relation seeds used to bootstrap glossary learning in the four domains for the English language.

3.2 Seed Selection

For each domain and language we manually selected five seed hypernymy relations, shown for the English language in Table 3. The seeds were selected by the authors on the basis of just two conditions: i) the seeds should cover different aspects of the domain and should, indeed, identify the domain implicitly, ii) at least 10,000 results should be returned by the search engine when querying it with the seeds plus the `glossaryKeyword` (see step (2) of GlossBoot). The seed selection was not fine-tuned (i.e., it was not adjusted to improve performance), so it might well be that better seeds would provide better results (see, e.g., (Kozareva and Hovy, 2010b)). However, this type of consideration is beyond the scope of this paper.

3.2.1 Evaluation measures

We performed experiments to evaluate the quality of both terms and glosses, as jointly extracted by GlossBoot.

Terms. For each domain and language we calculated coverage, extra-coverage and precision of the acquired terms T. Coverage is the ratio of extracted terms in T also contained in the gold standard $\hat{T}$ to the size of $\hat{T}$. Extra-coverage is calculated as the ratio of the additional extracted terms in $T \setminus \hat{T}$ over the number of gold standard terms $\hat{T}$. Finally, precision is the ratio of extracted terms in T deemed to be within the domain. To calculate precision we randomly sampled 5% of the retrieved terms and asked two human annotators to manually tag their domain pertinence (with adjudication in case of disagreement; $\kappa = .62$, indicating substantial agreement). Note that by sampling on the entire set T, we calculate the precision of both terms in $T \cap \hat{T}$, i.e., in the gold standard, and terms in $T \setminus \hat{T}$, i.e., not in the gold standard, which are not necessarily outside the domain.

Glosses. We calculated the precision of the extracted glosses as the ratio of glosses which were both well-formed textual definitions and specific to the target domain. Precision was determined on a random sample of 5% of the acquired glosses for each domain and language. The annotation was made by two annotators, with $\kappa = .675$, indicating substantial agreement.

			Botany	Comput.	Environ.	Finance
EN	Gold std.	terms	772	421	713	1777
	GlossBoot	terms	5598	3738	4120	5294
		glosses	11663	4245	5127	6703
FR	Gold std.	terms	662	278	117	109
	GlossBoot	terms	3450	3462	1941	1486
		glosses	5649	3812	2095	1692
IT	Gold std.	terms	205	244	450	441
	GlossBoot	terms	1965	3356	1630	3601
		glosses	2678	5891	1759	5276

Table 4: Size of the gold-standard and automatically-acquired glossaries for the four domains in the three languages of interest.

3.3 Parameter tuning

We tuned the minimum and maximum length of both p_L and p_R (see step (3) of GlossBoot) and the threshold θ that we use to filter out non-domain glosses (see step (4) of GlossBoot) using an extra domain, i.e., the Arts domain. To do this, we created a development dataset made up of the full set of 394 terms from the Tate Gallery glossary,[10] and bootstrapped our glossary extraction method with just one seed, i.e., (*fresco*, *painting*). We chose an optimal value of $\theta = 0.1$ on the basis of a harmonic mean of coverage and precision. Note that, since precision also concerns terms not in the gold standard, we had to manually validate a sample of the extracted terms for each of the 21 tested values of $\theta \in \{0, 0.05, 0.1, \ldots, 1.0\}$.

4 Results and Discussion

4.1 Terms

The size of the extracted terminologies for the four domains after five iterations are reported in Table 4. In Table 5 we show examples of the possible scenarios for terms: in-domain extracted terms

[10] http://www.tate.org.uk/collections/glossary/

	In-domain (in gold std, $\in \hat{T} \cap T$)	In-domain (not in gold std, $\in T \setminus \hat{T}$)	Out-of-domain (not in gold std, $\in T \setminus \hat{T}$)	In-domain (missed, $\in \hat{T} \setminus T$)
Computing	software, inheritance, microprocessor	clipboard, even parity, sudoer	gs1-128 label, grayscale, quantum dots	openwindows, sun microsystems, hardwired
Botany	pollinium, stigma, spore	vegetation, dichogamous, fertilisation	ion, free radicals, mana-mana	nomenclature, endemism, insectivorous
Environment	carcinogen, footprint, solar power	frigid soil, biosafety, fire simulator	epidermis, science park, alum	g8, best practice, polystyrene
Finance	cash, bond, portfolio	trustor, naked option, market price	precedent, immigration, heavy industry	co-location, petrodollars, euronext

Table 5: Examples of extracted (and missed) terms.

		Botany	Comput.	Environ.	Finance
EN	Precision	95%	98%	94%	98%
	Coverage	85%	40%	35%	32%
	Extra-coverage	640%	848%	542%	266%
FR	Precision	80%	97%	83%	98%
	Coverage	97%	27%	14%	26%
	Extra-coverage	425%	1219%	1646%	1350%
IT	Precision	89%	98%	76%	99%
	Coverage	42%	27%	11%	73%
	Extra-coverage	511%	1349%	356%	746%

Table 6: Precision, coverage and extra-coverage of the term extraction phase after 5 iterations.

which are also found in the gold standard (column 2), in-domain extracted terms but not in the gold standard (column 3), out-of-domain extracted terms (column 4), and domain terms in the gold standard but not extracted by our approach (column 5).

A quantitative evaluation is provided in Table 6, which shows the percentage results in terms of precision, coverage, and extra-coverage after 5 iterations of GlossBoot. For the English language we observe good coverage (between 32% and 40% on three domains, with a high peak of 85% coverage on Botany) and generally very high precision values. Moreover for the French and the Italian languages we observe a peak in the Botany and Finance domains respectively, while the lowest performances in terms of precision and coverage are observed for Environment, i.e., the most interdisciplinary domain.

In all three languages GlossBoot provides very high extra coverage of domain terms, i.e., additional terms which are not in the gold standard but are returned by our system. The figures, shown in Table 6, range between 266% (4726/1777) for the English Finance domain and 1646% (1926/117) for the French Environment domain. These results, together with the generally high precision values, indicate the larger extent of our bootstrapped glossaries compared to our gold standards.

Botany		Computing		Environm.		Finance	
Min	Max	Min	Max	Min	Max	Min	Max
26%	68%	8%	39%	5%	33%	14%	30%

Table 7: Coverage ranges for single-seed term extraction for the English language.

Number of seeds. Although the choice of selecting five hypernymy relation seeds is quite arbitrary, it shows that we can acquire a reliable terminology with minimal human intervention. Now, an obvious question arises: what if we bootstrapped GlossBoot with fewer hypernym seeds, e.g., just one seed? To answer this question we replicated our English experiments on each single (term, hypernym) pair in our seed set. In Table 7 we show the coverage ranges – i.e., the minimum and maximum coverage values – for the five seeds on each domain. We observe that the maximum coverage can attain values very close to those obtained with five seeds. However, the minimum coverage values are much lower. So, if we adopt a 1-seed bootstrapping policy there is a high risk of acquiring a poorer terminology unless we select the single seed very carefully, whereas we have shown that just a few seeds can cope with domain variability. Similar considerations can be made regarding different seed set sizes (we also tried 2, 3 and 4). So five is not a magic number, just one which can guarantee an adequate coverage of the domain.

Number of iterations. In order to study the coverage trend over iterations we selected 5 seeds for our tuning domain (i.e., Arts, see Section 3.3). Figure 3 shows the size (left graph), coverage, extra-coverage and precision (middle graph) of the acquired glossary after each iteration, from 1 to 20. As expected, (extra-)coverage grows over iterations, while precision drops. Stopping at iteration 5, as we do, is optimal in terms of the harmonic mean of precision and coverage (right graph in Figure 3).

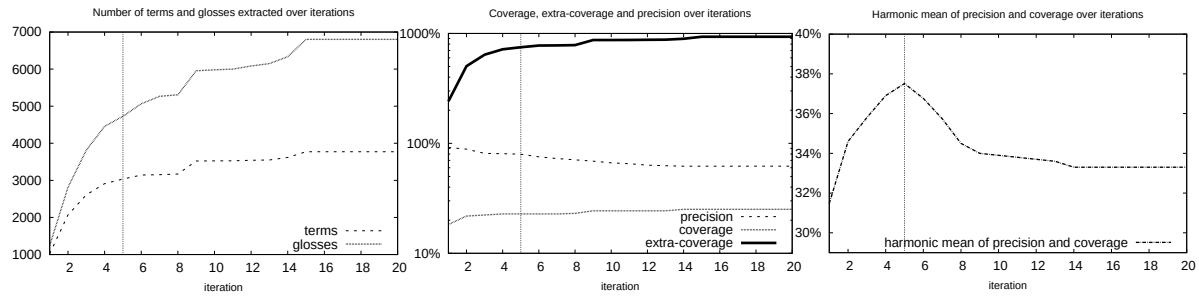

Figure 3: Size, coverage and precision trends for Arts (tuning domain) over 20 iterations for English.

	Botany	Comput.	Environm.	Finance
EN	96%	94%	97%	97%
FR	88%	89%	88%	95%
IT	94%	98%	83%	99%

Table 8: Precision of the glosses for the four domains and for the three languages.

4.2 Glosses

We show the results of gloss evaluation in Table 8. Precision ranges between 83% and 99%, with three domains performing above 92% on average across languages, and the Environment domain performing relatively worse because of its highly interdisciplinary nature (89% on average). We observe that these results are strongly correlated with the precision of the extracted terms (cf. Table 6), because the retrieved glosses of domain terms are usually in-domain too, and follow a definitional style because they come from glossaries. Note, however, that the gloss precision can also be higher than term precision, because many pertinent glosses might be extracted for the same term, cf. Table 4.

5 Comparative Evaluation

5.1 Comparison with Google Define

We performed a comparison with Google Define,[11] a state-of-the-art definition search service. This service inputs a term query and outputs a list of glosses. First, we randomly sampled 100 terms from our gold standard for each domain and each of the three languages. Next, for each domain and language, we manually calculated the fraction of terms for which an in-domain definition was provided by Google Define and GlossBoot. Table 9 shows the coverage results.

Google Define outperforms our system on all four domains (with a few exceptions). However we note that Google Define: i) requires knowing the domain term to be defined in advance, whereas we jointly acquire thousands of terms and glosses starting from just a few seeds; ii) does not discriminate between glosses pertaining to the target domain and glosses concerning other fields or senses, whereas we extract domain-specific glosses.

		Botany	Comput.	Environm.	Finance
EN	Google Define	90%	87%	84%	82%
	GlossBoot	77%	47%	44%	51%
FR	Google Define	40%	48%	36%	82%
	GlossBoot	88%	42%	22%	32%
IT	Google Define	52%	74%	78%	80%
	GlossBoot	64%	38%	44%	92%

Table 9: Number of domain glosses (from a random sample of 100 gold standard terms per domain) retrieved using Google Define and GlossBoot.

5.2 Comparison with TaxoLearn

We also compared GlossBoot with a recent approach to glossary learning embedded into a framework for graph-based taxonomy learning from scratch, called TaxoLearn (Navigli et al., 2011). Since this approach requires the manual selection of a domain corpus to automatically extract terms and glosses, we decided to keep a level playing field and experimented with the same domain used by the authors, i.e., Artificial Intelligence (AI). TaxoLearn was applied to the entire set of IJCAI 2009 proceedings, resulting in the extraction of 427 terms and 834 glosses.[12] As regards GlossBoot, we selected 10 seeds to cover all the fields of AI, obtaining 5827 terms and 6716 glosses after 5 iterations, one order of magnitude greater than TaxoLearn.

As for the precision of the extracted terms, we randomly sampled 50% of them for each system. We show in Table 10 (first row) the estimated term

[11] Accessible from Google search by means of the `define:` keyword.

[12] Available at: http://lcl.uniroma1.it/taxolearn

	GlossBoot	TaxoLearn
Term Precision	82.3% (2398/2913)	77.0% (164/213)
Gloss Precision	82.8% (2780/3358)	78.9% (329/417)

Table 10: Estimated term and gloss precision of GlossBoot and TaxoLearn for the Artificial Intelligence domain.

precision for GlossBoot and TaxoLearn. The precision value for GlossBoot is lower than the precision values of the four domains in Table 6, due to the AI domain being highly interdisciplinary. TaxoLearn obtained a lower precision because it acquires a full-fledged taxonomy for the domain, thus also including higher-level concepts which do not necessarily pertain to the domain.

We performed a similar evaluation for the precision of the acquired glosses, by randomly sampling 50% of them for each system. We show in Table 10 (second row) the estimated gloss precision of GlossBoot and TaxoLearn. Again, GlossBoot outperforms TaxoLearn, retrieving a larger amount of glosses (6716 vs. 834) with higher precision. We remark, however, that in TaxoLearn glossary extraction is a by-product of the taxonomy learning process.

Finally, we note that we cannot compare with approaches based on lexical patterns (such as (Kozareva and Hovy, 2010a)), because they are not aimed at learning glossaries, but just at retrieving sentence snippets which contain pairs of terms/hypernyms (e.g., "*supervised systems* such as *decision trees*").

6 Related Work

There are several techniques in the literature for the automated acquisition of definitional knowledge. Fujii and Ishikawa (2000) use an n-gram model to determine the definitional nature of text fragments, whereas Klavans and Muresan (2001) apply pattern matching techniques at the lexical level guided by cue phrases such as "is called" and "is defined as". Cafarella et al. (2005) developed a Web search engine which handles more general and complex patterns like "cities such as $ProperNoun(Head(NP))$" in which it is possible to constrain the results with syntactic properties. More recently, a domain-independent supervised approach was presented which learns Word-Class Lattices (WCLs), i.e. lattice-based definition classifiers that are applied to candidate sentences containing the input terms (Navigli and Velardi, 2010). WCLs have been shown to perform with high precision in several domains (Velardi et al., 2013).

To avoid the burden of manually creating a training dataset, definitional patterns can be extracted automatically. Reiplinger et al. (2012) experimented with two different approaches for the acquisition of lexical-syntactic patterns. The first approach involves bootstrapping patterns from a domain corpus, and then manually refining the acquired patterns. The second approach, instead, involves automatically acquiring definitional sentences by using a more sophisticated syntactic and semantic processing. The results shows high precision in both cases.

However, these approaches to glossary learning extract unrestricted textual definitions from open text. In order to filter out non-domain definitions, Velardi et al. (2008) automatically extract a domain terminology from an input corpus which they later use for assigning a domain score to each harvested definition and filtering out non-domain candidates. The extraction of domain terms from corpora can be performed either by means of statistical measures such as specificity and cohesion (Park et al., 2002), or just TF*IDF (Kim et al., 2009).

To avoid the use of a large domain corpus, terminologies can be obtained from the Web by using Doubly-Anchored Patterns (DAPs) which, given a (term, hypernym) pair, harvest sentences matching manually-defined patterns like "<hypernym> such as <term>, and *" (Kozareva et al., 2008). Kozareva and Hovy (2010a) further extend this term extraction process by harvesting new hypernyms using the corresponding inverse patterns (called DAP^{-1}) like "* such as <term$_1$>, and <term$_2$>". Similarly to our approach, they drop the requirement of a domain corpus and start from a small number of (term, hypernym) seeds. However, while Doubly-Anchored Patterns have proven useful in the induction of domain taxonomies (Kozareva and Hovy, 2010a), they cannot be applied to the glossary learning task, because the extracted sentences are not formal definitions.

In contrast, GlossBoot performs the novel task of multilingual glossary learning from the Web by bootstrapping the extraction process with a few (term, hypernym) seeds. Bootstrapping techniques (Brin, 1998; Agichtein and Gravano, 2000; Paşca et al., 2006) have been successfully applied to several tasks, including high-precision semantic lexicon extraction from large corpora (Riloff and Jones, 1999; Thelen and Riloff, 2002; McIntosh

	Domain	Term	Gloss
EN	Botany	deciduous	losing foliage at the end of the growing season.
	Computing	information space	The abstract concept of everything accessible using networks: the Web.
	Finance	discount	The difference between the lower price paid for a security and the security's face amount at issue.
FR	Botany	insectivore	Qui capture des insectes et en absorbe les matières nutritives.
	Computing	notebook	C'est l'appellation d'un petit portable d'une taille proche d'une feuille A4.
	Environment	écosystème	Ensemble des êtres vivants et des éléments non vivants d'un milieu qui sont liés vitalement entre eux.
IT	Computing	link	Collegamento tra diverse pagine web, può essere costituito da immagini o testo.
	Environment	effetto serra	Riscaldamento dell'atmosfera terrestre dovuto alla presenza di gas nell'atmosfera (anidride carbonica, metano e vapore acqueo) che ostacolano l'uscita delle radiazioni infrarosse emesse dal suolo terreste verso l'alto.
	Finance	spread	Indica la differenza tra la quotazione di acquisto e quella di vendita.

Table 11: An excerpt of the domain glossaries acquired for the three languages.

and Curran, 2008; McIntosh and Curran, 2009), learning semantic relations (Pantel and Pennacchiotti, 2006), extracting surface text patterns for open-domain question answering (Ravichandran and Hovy, 2002), semantic tagging (Huang and Riloff, 2010) and unsupervised Word Sense Disambiguation (Yarowsky, 1995). By exploiting the (term, hypernym) seeds to bootstrap the iterative acquisition of extraction patterns from Web glossary pages, we can cover the high variability of textual definitions, including both sentences matching the above-mentioned lexico-syntactic patterns (e.g., "a corpus is a collection of documents") and glossary-style definitions (e.g., "corpus: a collection of document") independently of the target domain and language.

7 Conclusions

In this paper we have presented GlossBoot, a new, minimally-supervised approach to multilingual glossary learning. Starting from a few hypernymy relation seeds which implicitly identify the domain of interest, we apply a bootstrapping approach which iteratively obtains HTML patterns from Web glossaries and then applies them to the extraction of term/gloss pairs. To our knowledge, GlossBoot is the first approach to large-scale glossary learning which jointly acquires thousands of terms and glosses for a target domain and language with minimal supervision.

The gist of GlossBoot is our glossary bootstrapping approach, thanks to which we can drop the requirements of existing techniques such as the availability of domain text corpora, which often do not contain enough definitions, and the manual specification of lexical patterns, which typically extract sentence snippets, instead of formal glosses.

GlossBoot will be made available to the research community as open-source software. Beyond the immediate usability of its output and its effective use for domain Word Sense Disambiguation (Faralli and Navigli, 2012), we wish to show the benefit of GlossBoot in gloss-driven approaches to ontology learning (Navigli et al., 2011; Velardi et al., 2013) and semantic network enrichment (Navigli and Ponzetto, 2012). In Table 11 we show an excerpt of the acquired glossaries. All the glossaries and gold standards created for our experiments are available from the authors' Web site `http://lcl.uniroma1.it/glossboot/`.

We remark that the terminologies covered with GlossBoot are not only precise, but also one order of magnitude greater than those covered in individual online glossaries. As future work we plan to study the ability of GlossBoot to acquire domain glossaries at different levels of specificity (i.e., domains vs. subdomains). We also plan to exploit the acquired HTML patterns for implementing an open-source glossary crawler, along the lines of Google Define.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the 5th ACM conference on Digital Libraries*, pages 85–94, San Antonio, Texas, USA.

Sergey Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings of the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK.

Michael J. Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. 2005. KnowItNow: Fast, scalable information extraction from the web. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 563–570, Vancouver, British Columbia, Canada.

Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2007. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems*, 25(2):1–30.

Weisi Duan and Alexander Yates. 2010. Extracting glosses to disambiguate word senses. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 627–635, Los Angeles, CA, USA.

Stefano Faralli and Roberto Navigli. 2012. A New Minimally-supervised Framework for Domain Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422, Jeju, Korea.

Atsushi Fujii and Tetsuya Ishikawa. 2000. Utilizing the World Wide Web as an encyclopedia: extracting term descriptions from semi-structured texts. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 488–495, Hong Kong.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 539–545, Nantes, France.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

Ruihong Huang and Ellen Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 275–285, Uppsala, Sweden.

Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. 2009. An unsupervised approach to domain-specific term extraction. In *Proceedings of the Australasian Language Technology Workshop*, pages 94–98, Sydney, Australia.

Judith Klavans and Smaranda Muresan. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of the American Medical Informatics Association (AMIA) Symposium*, pages 324–328, Washington, D.C., USA.

Zornitsa Kozareva and Eduard Hovy. 2010a. A semi-supervised method to learn and construct taxonomies using the Web. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1110–1118, Cambridge, MA, USA.

Zornitsa Kozareva and Eduard H. Hovy. 2010b. Not all seeds are equal: Measuring the quality of text mining seeds. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 618–626, Los Angeles, California, USA.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the Web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 1048–1056, Columbus, Ohio, USA.

Tara McIntosh and James R. Curran. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 97–105, CSIRO ICT Centre, Tasmania.

Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 396–404, Suntec, Singapore.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 1872–1877, Barcelona, Spain.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 809–816, Sydney, Australia.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), Sydney, Australia*, pages 113–120, Sydney, Australia.

Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. 2002. Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47, Philadelphia, Pennsylvania.

Melanie Reiplinger, Ulrich Schäfer, and Magdalena Wolska. 2012. Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 55–65, Jeju Island, Korea.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference*, pages 474–479, Menlo Park, CA, USA.

Horacio Saggion. 2004. Identifying definitions in text collections for question answering. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 1927–1930, Lisbon, Portugal.

Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 214–221, Salt Lake City, UT, USA.

Paola Velardi, Roberto Navigli, and Pierluigi D'Amadio. 2008. Mining the Web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3).

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation rivaling supervised methods. In *Proceedings of the* 33^{rd} *Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA, USA.