Growing Multi-Domain Glossaries from a Few Seeds using Probabilistic Topic Models

Stefano Faralli and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
{faralli,navigli}@di.uniroma1.it

Abstract

In this paper we present a minimally-supervised approach to the multi-domain acquisition of wide-coverage glossaries. We start from a small number of hypernymy relation seeds and bootstrap glossaries from the Web for dozens of domains using Probabilistic Topic Models. Our experiments show that we are able to extract high-precision glossaries comprising thousands of terms and definitions.

1 Introduction

Dictionaries, thesauri and glossaries are useful sources of information for students, scholars and everyday readers, who use them to look up words of which they either do not know, or have forgotten, the meaning. With the advent of the Web an increasing number of dictionaries and technical glossaries has been made available online, thereby speeding up the definition search process. However, finding definitions is not always immediate, especially if the target term pertains to a specialized domain. Indeed, not even well-known services such as Google Define are able to provide definitions for scientific or technical terms such as *taxonomy learning* or *distant supervision* in AI or *figure-four leglock* and *suspended surfboard* in wrestling.

Domain-specific knowledge of a definitional nature is not only useful for humans, it is also useful for machines (Hovy et al., 2013). Examples include Natural Language Processing tasks such as Question Answering (Cui et al., 2007), Word Sense Disambiguation (Duan and Yates, 2010; Faralli and Navigli, 2012) and ontology learning (Velardi et al., 2013). Unfortunately, most of the Web dictionaries and glossaries available online comprise just a few hundred definitions, and they therefore provide only a partial view of a domain. This is also the case with manually compiled glossaries created by means of collaborative efforts, such as Wikipedia.[1] The coverage issue is addressed by online aggregation services such as Google Define, which bring together definitions from several online dictionaries. However, these services do not classify textual definitions by domain: they just present the collected definitions for all the possible meanings of a given term.

In order to automatically obtain large domain glossaries, in recent years computational approaches have been developed which extract textual definitions from corpora (Navigli and Velardi, 2010; Reiplinger et al., 2012) or the Web (Velardi et al., 2008; Fujii and Ishikawa, 2000). The methods involving corpora 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

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

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 170–181,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

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.

The goal of the new approach outlined in this paper is to enable the automatic harvesting of large-scale, full-fledged domain glossaries for dozens of domains, an outcome which should be very useful for both human activities and automatic tasks. We present ProToDoG (Probabilistic Topics for multi-Domain Glossaries), a framework for growing multi-domain glossaries which has three main novelties:

i) **minimal human supervision:** a small set of hypernymy relation seeds for each domain is used to bootstrap the multi-domain acquisition process;

ii) **jointness:** our approach harvests terms and glosses at the same time;

iii) **probabilistic topic models** are leveraged for a simultaneous, high-precision multi-domain classification of the extracted definitions, with substantial performance improvements over our previous work on glossary bootstrapping, i.e., GlossBoot (De Benedictis et al., 2013).

ProToDog is able to harvest definitions from the Web and thus drop the requirement of large corpora for each domain. Moreover, apart from the need to select a few seeds, it avoids the use of training data or manually defined sets of lexical patterns. It is thus applicable to virtually any language of interest.

2 ProToDoG

Given a set of domains $D = \{d_1, ..., d_n\}$, for each domain $d \in D$ ProToDoG harvests a domain glossary G_d containing pairs of the kind (t, g) where t is a domain term and g is its textual definition, i.e., gloss. We show the pseudocode of ProToDoG in Algorithm 1.

Step 1. Initial seed selection: Algorithm 1 takes as input a set of domains D and, for each domain $d \in D$, a small set of hypernymy relation seeds $S_d = \{(t_1, h_1), \ldots, (t_{|S_d|}, h_{|S_d|})\}$, where the seed pair ($t_j$, h_j) contains a term t_j and its generalization h_j (e.g., (*linux*, *operating system*)). This is the only human input to the entire glossary acquisition 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 them. 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 out-of-domain glossaries of units and measures).

Algorithm 1 ProToDoG

```
Input:  the set of domains $D$,
        a set $S_d$ of hypernymy seeds for each domain
        $d \in D$
Output: a multi-domain glossary $G$
 1: $k \leftarrow 1$
 2: repeat
 3:     for each domain $d \in D$ do
 4:         $G_d^k \leftarrow \emptyset$
 5:         for each seed $(t_j, h_j) \in S_d$ do
 6:             $pages \leftarrow webSearch(t_j, h_j, \text{``}glossary\text{''})$
 7:             $G_d^k \leftarrow G_d^k \cup extractGlossary(pages)$
 8:         end for
 9:     end for
10:     create a topic model using glossaries from previous iterations
11:     infer topic assignments for iteration-$k$ glosses
12:     filter out non-domain glosses for each domain
13:     for each $d \in D$ do
14:         $S_d \leftarrow seedSelectionForNextIteration(G_d^k)$
15:     end for
16:     $k \leftarrow k + 1$
17: until $k > max$
18: for each domain $d \in D$ do
19:     recover filtered glosses into $G_d^{max+1}$
20:     $G_d \leftarrow \bigcup_{j=1,\ldots,max+1} G_d^j$
21: end for
22: return $G = \{(G_d, d) : d \in D\}$
```

The algorithm first sets the iteration counter k to 1 (line 1) and starts the first iteration of the glossary bootstrapping process (lines 2-17), each involving steps 2-4, described below. After each iteration k, for each domain d we keep track of the set of glosses G_d^k acquired during that iteration. After the last iteration, we perform step (5) of gloss recovery (lines 18-21).

Step 2. Web search and glossary extraction (lines 3-9): For each domain d, we first initialize the domain glossary for iteration k: $G_d^k := \emptyset$ (line 4). Then, for each seed pair $(t_j, h_j) \in S_d$, we submit the following query to a Web search engine: "t_j" "h_j" `glossary` and collect the top-ranking results for each query (line 6).[2] Each resulting page is a candidate glossary for the domain d.

We then call the $extractGlossary$ function (line 7) which extracts terms and glosses from the retrieved pages as follows. From each candidate page, we harvest all the text snippets s starting with t_j and ending with h_j (e.g., "*linux*</b> – an <i>*operating system*"), i.e., $s = t_j \ldots h_j$. For each such text snippet s, we extract the following pattern instance:

$$p_L \; t_j \; p_M \; gloss_s(t_j) \; p_R,$$

where:

- p_M is the longest sequence of HTML tags and non-alphanumeric characters between t_j and the glossary definition (e.g., "</b> –" between "linux" and "an" in the above example);
- $gloss_s(t_j)$ is the gloss of t_j obtained by moving to the right of p_M until we reach a non-formatting tag element (e.g., <span>, <p>, <div>), while ignoring formatting elements such as <b>, <i> and <a> which are typically included within a definition sentence;
- p_L and p_R are the longest sequences of HTML tags on the left of t_j and the right of $gloss_s(t_j)$, respectively.

For instance, given the HTML snippet "…<p><b>linux</b> – an <i>operating system</i> developed by Linus Torvalds</p>…" we extract the following pattern instance: p_L = "<p><b>", t_j = "*linux*", p_M = "</b> –", $gloss_s(t_j)$ = "an <i>*operating system*</i> developed by Linus Torvalds", p_R ="</p>".

Then we generalize the above pattern instance by replacing t_j and $gloss_s(t_j)$ with *, obtaining:

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

For the above example, we obtain the following pattern:

<p><b> * </b> – * </p>.

We add the first sentence of the retrieved gloss $gloss_s(t_j)$ to our glossary G_d^k, i.e., $G_d^k := G_d^k \cup \{(t_j, first(gloss_s(t_j)))\}$, where $first(g)$ returns the first sentence of gloss g. 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 $p_L * p_M * p_R$, and adding them to G_d^k.

As a result of step (2), for each domain $d \in D$ we obtain a glossary G_d^k for the terms discovered at iteration k.

Step 3. Topic modeling and gloss filtering (lines 10-12): Unfortunately, not all (term, gloss) pairs in a glossary G_d^k will pertain to the domain d. For instance, we might end up retrieving interdisciplinary or even unrelated glossaries. In order to address this fuzziness, we model domains with a Probabilistic Topic Model (PTM) (Blei et al., 2003; Steyvers and Griffiths, 2007). PTMs model a given text document as a mixture of topics. In our case topics are domains and we, first, create a topic model from the domain glossaries acquired before the current iteration k, then, second, use the topic model to estimate the domain assignment of each new pair (term, gloss) in our glossaries G_d^k, i.e., obtained at iteration k, third, filter out non-domain glosses.

Creating the topic model (line 10): For a given iteration k and domain d, we first define the terminology accumulated up until iteration $k-1$ for that domain as the set $T_d^{1,k-1} := \bigcup_{j=1}^{k-1} T_d^j$, where T_d^j is the set of terms acquired at iteration j, i.e., $T_d^j := \{t : \exists(t, g) \in G_d^j\}$.[3] Then we define:

- $W := \bigcup_{d \in D} T_d^{1,k-1}$ as the entire terminology acquired up until iteration $k-1$ for all domains, i.e., the full set of terms independently of their domain;
- $M := \bigcup_{d \in D} \bigcup_{j=1}^{k-1} G_d^j$ as the multi-domain glossary acquired up until iteration $k-1$, i.e., the full set of pairs (term, gloss) independently of their domain;[4]

[2] We use the Google Ajax API, which returns the 64 top-ranking search results.

[3] For the first iteration, i.e., when $k = 1$, we define $T_d^{1,0} := \{t : \exists(t, g) \in G_d^1\}$, i.e., we use the terminology resulting from step (2) of the first iteration.

[4] For $k = 1$, $M := \bigcup_{d \in D} G_d^1$.

- Two count matrices, i.e., the word-domain matrix C^{WD} and the gloss-domain matrix C^{MD}, such that: $C^{WD}_{w,d}$ counts the number of times $w \in W$ is assigned to domain $d \in D$, i.e., it occurs in the glosses of domain d; $C^{MD}_{(t,g),d}$ counts the number of words in g assigned to domain d.

At this point, as shown by Steyvers and Griffiths (2007), we can estimate the probability $\phi^{(d)}_w$ for word w, and the probability $\theta^{(t,g)}_d$ for a term/gloss pair (t, g), of belonging to domain d:

$$\phi^{(d)}_w = \frac{C^{WD}_{w,d} + \beta}{\sum_{w'=1}^{|W|} C^{WD}_{w',d} + |W|\beta}; \quad (1)$$

$$\theta^{(t,g)}_d = \frac{C^{MD}_{(t,g),d} + \alpha}{\sum_{d'=1}^{|D|} C^{MD}_{(t,g),d'} + |D|\alpha} \quad (2)$$

where α and β are smoothing factors.[5] The two above probabilities represent the core of our topic model of the domain knowledge acquired up until iteration $k - 1$.

Probabilistic modeling of iteration-k glosses (line 11): We now utilize the above topic model to estimate the probabilities in Formulas 1 and 2 for the newly acquired glosses at iteration k. To this end we define $M' := \bigcup_{d \in D} G^k_d$ as the union of the (term, gloss) pairs at iteration k and $W' := \bigcup_{d \in D} T^k_d \bigcap W$ as the union of terms acquired at iteration k, but also occurring in W (i.e., the entire terminology up until iteration $k - 1$). Then we apply Gibbs sampling (Blei et al., 2003; Phan et al., 2008) to estimate the probability of each pair $(t, g) \in M'$ of pertaining to a domain d by computing:

$$\theta^{'(t,g)}_d = \frac{R^{M'D}_{(t,g),d} + \alpha}{\sum_{d'=1}^{|D|} R^{M'D}_{(t,g),d'} + |D|\alpha} \quad (3)$$

where the gloss-domain matrix $R^{M'D}$ is initially defined by counting random domain assignments for each word w' in the bag of words of each (term, gloss) pair $\in M'$. Next, the domain assignment counts in $R^{M'D}$ are iteratively updated using Gibbs sampling.[6]

Filtering out non-domain glosses (line 12): Now, for each domain $d \in D$, for each pair $(t, g) \in G^k_d$ we have a probability $\theta^{'(t,g)}_d$ of belonging to d. We mark (t, g) as a non-domain item if $\theta^{'(t,g)}_d < \delta$, where δ is a confidence threshold, or if $\theta^{'(t,g)}_d$ is not maximum among all domains in D. Non-domain pairs are removed from G^k_d and stored into a set A_d for possible recovery after the last iteration (see step (5)).

Step 4. Seed selection for next iteration (lines 13-15): For each domain $d \in D$, we now select the new set of hypernymy relation seeds to be used to start the next iteration. First, for each newly-acquired term/gloss pair $(t, g) \in G^k_d$, we automatically extract a candidate hypernym h from the textual gloss g. To do this we use a simple heuristic which just selects the first content term in the gloss.[7] Then we sort all the glosses in G^k_d 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 $\theta^{'(t,g)}_d$. Finally we select the (term, hypernym) pairs corresponding to the $|S_d|$ top-ranking glosses as the new set of seeds for the next iteration.

Next, we increment k (line 16 of Algorithm 1) and if the maximum number of iterations is reached we jump to step (5). Otherwise, we go back to step (2) of our glossary bootstrapping algorithm with the new set of seeds S_d.

Step 5. Gloss recovery (line 19): After all iterations, the entire multi-domain terminology W (cf. step (3)) may contain several new terms which were not present when a given gloss g was filtered out. So, thanks to the last-iteration topic model, the gloss g might come back into play because its words are now important cues for a domain. To reassess the domain pertinence of (term, gloss) pairs in A_d for each d, we just reapply the entire step (3) by setting $G^{max+1}_d := A_d$ for each $d \in D$. As a result, we

[5] As experienced by Steyvers and Griffiths (2007), the values of $\alpha = 50/|D|$ and $\beta = 0.01$ work well with many different text collections.

[6] For the PTM part of ProToDoG we used the JGibbLDA library, a Java Implementation of Latent Dirichlet Allocation (LDA) using Gibbs Sampling for Parameter Estimation and Inference, available at: http://jgibblda.sourceforge.net/

[7] While more complex strategies could be devised, e.g., lattice-based hypernym extraction (Navigli and Velardi, 2010), we found that this heuristic works well because, even when it is not a hypernym, the first term acts as a cue word for the defined term.

obtain an updated glossary G_d^{max+1} which contains all the recovered glosses.

Final output: For each domain $d \in D$ the final output of ProToDoG is a domain glossary $G_d := \bigcup_{j=1,\dots,max+1} G_d^j$. Finally the algorithm aggregates all glossaries G_d into a multi-domain glossary G (line 22).

3 Experimental Setup

3.1 Domains

For our experiments we selected 30 different domains ranging from Arts to Warfare, mostly following the domain classification of Wikipedia featured articles (full list at `http://lcl.uniroma1.it/protodog`). The set includes several technical domains, such as Chemistry, Geology, Meteorology, Mathematics, some of which are highly interdisciplinary. For instance, the Environment domain covers terms from fields such as Chemistry, Biology, Law, Politics, etc.

3.2 Gold Standard

Since our evaluations required considerable human effort, in what follows we calculated all performances on a random set of 10 domains, shown in the top row of Table 1. For each of these 10 domains we selected well-reputed glossaries on the Web as gold standards, including the Reuters glossary of finance, the Utah computing glossary and many others (full list at the above URL). We show the size of our 10 gold-standard datasets in Table 1.

3.3 Evaluation measures

We evaluated the quality of both terms and glosses, as jointly extracted by ProToDoG.

3.3.1 Terms

For each domain 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 $\hat{T}$ over 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 randomly 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, but which are not necessarily outside the domain.

3.3.2 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. The annotation was made by two annotators, with $\kappa = .675$, indicating substantial agreement. The annotators were provided with specific guidelines available on the ProToDoG Web site (see URL above).

3.4 Comparison

We compared ProToDog against:

- **BoW**: a bag-of-words variant in which step (3) is replaced by a simple bag-of-words scoring approach which assigns a score to each term/gloss pair $(t, g) \in G_d^k$ as follows:

$$score(g) = \frac{|Bag(g) \cap T_d^{1,k-1}|}{|Bag(g)|}. \qquad (4)$$

  where $Bag(g)$ contains all content words in g. At iteration k, we filter out those glosses whose $score(g) < \sigma$, where σ is a threshold tuned in the same manner as δ (see Section 3.5). This approach essentially implements GlossBoot, our previous work on domain glossary bootstrapping (De Benedictis et al., 2013).

- **Wikipedia**: since Wikipedia is the largest collaborative resource, covering hundreds of fields of knowledge, we devised a simple heuristic for producing multi-domain glossaries from Wikipedia, so as to compare their performance against our gold standards. For each target domain we manually selected one

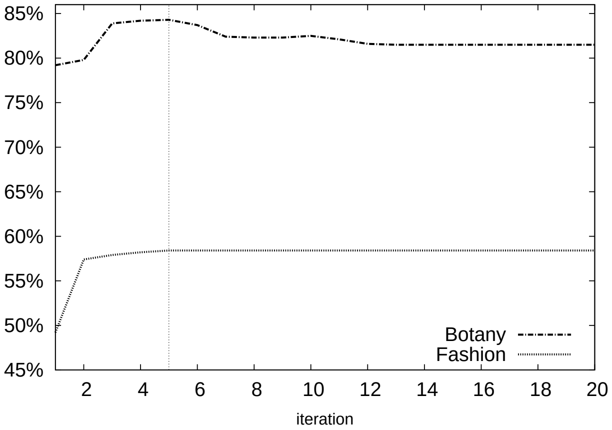

Figure 1: Harmonic mean of precision and coverage for Botany and Fashion (tuning domains) over 20 iterations ($|S_d|$=5, δ=0.03).

or more Wikipedia categories representing the domain (for instance, Category:Arts for Arts, Category:Business for Finance, etc.). Then, for each domain d, we picked out all the Wikipedia pages tagged either with the categories selected for d or their direct subcategories (e.g., Category:Creative works) or sub-subcategories (e.g., Category:Genres). From each page we extracted a (page title, gloss) pair, where the gloss was obtained by extracting the first sentence of the Wikipedia page, as done, e.g., in BabelNet (Navigli and Ponzetto, 2012). Since subcategories might have more parents and might thus belong to multiple domains, we discarded pages assigned to more than 2 domains.

3.5 Parameter tuning

In order to choose the optimal values of the parameters of ProToDoG (number $|S_d|$ of seeds per domain, number max of iterations, and filtering threshold δ) and BoW (σ threshold) we selected two extra domains, i.e., Botany and Fashion, not used in our tests, together with the corresponding gold standard Web glossaries.

As regards the number of seeds, we defined an initial pool of 10 seeds for each of the two tuning domains and studied the average performance of 5 random sets of x seeds (from the initial pool), when $x = 1, 3, 5, 7, 9$. As regards the number of iterations, we explored all values between 1 and 20. Finally, for the filtering thresholds δ and σ for ProToDoG PTM and its BoW variant, we tried values of $\delta \in \{0, 0.03, 0.06, \dots, 0.6\}$ and $\sigma \in \{0, 0.05, 0.1, \dots, 1.0\}$, respectively.

Given the high number of possible parameter value configurations, we first explored the entire search space automatically by calculating the coverage of ProToDoG PTM (and BoW) with each configuration against our tuning gold standards. Then we identified as optimal candidates those "frontier" configurations for which, when moving from a lower-coverage configuration, coverage reached a maximum. We then calculated the precision of each optimal candidate configuration by manually validating a 3% random sample of the resulting glossaries for the two tuning domains. The optimal configuration for ProToDoG was $|S_d| = 5$, $max = 5$, $\delta = 0.03$, while for BoW was $\sigma = 0.1$.

In Figure 1 we show the performance trend over iterations for our two tuning domains when $|S_d| = 5$ and $\delta = 0.03$. Performance is calculated as the harmonic mean of precision and coverage of the acquired glossary after each iteration, from 1 to 20. We can see that after 5 iterations performance decreases for Botany (a highly interdisciplinary domain) due to lower precision, while it remains stable for Fashion due to the lack of newly-acquired glosses.

3.6 Seed Selection

For each domain d we manually selected five seed hypernymy relations as the seed sets S_d input to Algorithm 1 (see Section 3.5). 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, indeed, should 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 glossary keyword (see line 6 of Algorithm 1). 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 (Kozareva and Hovy, 2010a)). However, such a study is beyond the scope of this paper.

		Art	Business	Chemistry	Computing	Environment	Food	Law	Music	Physics	Sport
Gold	t/g	394	1777	164	421	713	946	180	218	315	146
PTM	t	4253	7370	2493	3412	3009	1526	1836	1647	3847	1696
	g	7386	9795	3841	4186	3552	2175	4141	2729	5197	2938
BoW	t	4012	7639	1174	3127	3644	1827	1773	1166	4471	1990
	g	5923	8999	1414	3662	4334	2601	4024	1249	6956	3425
Wiki	t,g	107.1k	48.4k	8137	32.0k	23.6k	5698	13.5k	84.1k	33.8k	267.5k

Table 1: Size of the gold standard and the automatically-acquired glossaries for 10 of the 30 selected domains (t: number of terms, g: number of glosses).

4 Results and Discussion

4.1 Terms

The size of the extracted terminologies for the 10 domains after five iterations is reported in Table 1 (the output for all 30 domains is available at the above URL, cf. Section 3.1). ProToDoG PTM and its BoW variant extract thousands of terms and glosses for each domain, whereas the number of glosses obtained from Wikipedia (cf. Section 3.4) varies depending upon the domain, from thousands to hundreds of thousands. Note that there is no overlap between the glossaries extracted by ProToDoG and the set of Wikipedia articles, since the latter are not organized as glossaries.

In Table 2 we show the percentage results in terms of precision (P), coverage (C), and extra-coverage (X, see Section 3.3 for definitions) for ProToDoG PTM and its BoW variant and for the Wikipedia glossary. With the exception of the Food domain, ProToDoG achieves the best precision. The Wikipedia glossary has fluctuating precision values, ranging between 25% and 90%, due to the heterogeneous nature of subcategories. ProToDog achieves the best coverage of gold standard terms on 6 of the 10 domains, with the BoW variant obtaining slightly higher coverage on 3 domains and +10% on the Food domain. The coverage of Wikipedia glossaries, instead, with the sole exception of Sport, is much lower, despite the use of (sub)subcategories (cf. Section 3.4). Both ProToDoG PTM and BoW achieve very high extra-coverage percentages, meaning that they are able to go substantially beyond our domain gold standards, but it is the Wikipedia glossary which achieves the highest extra-coverage values. To get a better insight into the quality of extra-coverage we calculated the percentage of named entities (i.e., encyclopedic) among the terms extracted by each of the different approaches. Comparing results across the (E) columns of Table 2 it can be seen that high percentages of the terms extracted by Wikipedia are named entities, which is in marked contrast to the 0%-1% extracted by ProToDog. This is as should be expected for an encyclopedia, whose coverage focuses on people, places, brands, etc. rather than concepts.

To summarize, ProToDoG PTM outperforms both BoW and Wikipedia in terms of precision, while at the same time achieving both competitive coverage and extra-coverage. The Wikipedia glossary suffers from fluctuating precision values across domains and overly encyclopedic coverage of terms.

4.2 Glosses

We show the results of gloss evaluation in Table 2 (last two columns) for ProToDoG PTM and BoW (we do not report the precision values for Wikipedia, as they are slightly lower than those obtained for terms). Precision ranges between 89% and 99% for ProToDoG PTM and between 82% and 97% for BoW. We observe that these results are strongly correlated with the precision of the extracted terms (cf. Table 2), because the retrieved glosses of domain terms are usually in-domain too, and follow a definitional style since they come from glossaries. Note, however, that the gloss precision could also be

	terms												glosses	
	PTM				BoW				Wiki				PTM	BoW
	P	C	X	E	P	C	X	E	P	C	X	E	P	P
Art	**92**	**26**	1053	1	86	25	992	0	81	19	23.4k	67	**93**	87
Business	**95**	41	374	0	90	**43**	387	0	37	15	2692	31	**96**	91
Chemistry	**99**	**77**	1410	0	95	73	643	0	49	18	12.9k	3	**98**	96
Computing	**95**	**43**	767	0	93	40	702	0	81	30	7506	36	**96**	94
Environment	**91**	**29**	393	0	84	28	482	0	25	9	3302	12	**89**	82
Food	91	21	1404	0	**97**	**31**	1621	0	81	9	3997	25	92	**95**
Law	**98**	**89**	931	0	95	87	897	0	35	34	7406	16	**99**	97
Music	**94**	**98**	660	0	93	84	453	0	90	50	37.1k	84	**96**	95
Physics	**97**	43	1178	0	91	**46**	1373	0	68	25	10.6k	10	**95**	89
Sport	**98**	22	1139	1	96	**23**	1339	1	87	44	178.2k	83	**97**	96

Table 2: Precision (P), coverage (C), extra-coverage (X), encyclopedic (E) percentages after 5 iterations.

	Art	Business	Chemistry	Computing	Environment	Food	Law	Music	Physics	Sport
Google Define	76	80	93	86	88	91	96	96	98	84
ProToDoG	27	41	81	40	37	19	85	98	47	27

Table 3: Number of domain glosses (from a random sample of 100 gold standard terms per domain) retrieved using Google Define and ProToDoG.

higher than term precision, thanks to many pertinent glosses being extracted for the same term (cf. Table 1).

In Table 4 we show an excerpt of the multi-domain glossary extracted by ProToDoG for the Art, Business and Sport domains.

5 Comparative Evaluation

5.1 Comparison with Google Define

We performed a comparison with Google Define,[8] 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. Next, for each domain, we manually calculated the fraction of terms for which at least one in-domain definition was provided by Google Define and ProToDoG. Table 3 shows the coverage results. In this experiment, Google Define outperforms our system on 9 of the 10 analyzed domains. However, we note that when searching for domain-specific knowledge only, Google Define: i) needs to know the domain term to be defined in advance, while ProToDoG jointly acquires domain terms and glosses starting from just a few seeds; ii) does not discriminate between glosses pertaining to the target domain and glosses pertaining to other fields or senses, whereas ProToDog extracts terms and glosses specific to each domain of interest.

5.2 Comparison with TaxoLearn

We also compared ProToDoG with the output of a state-of-the-art taxonomy learning framework, called TaxoLearn (Navigli et al., 2011). We did this because i) TaxoLearn extracts terms and glosses from domain corpora in order to create a domain taxonomy; ii) it is one of the few systems which extracts both terms and glosses from specialized corpora; iii) the extracted glossaries are available online.[9] Therefore we compared the performance of ProToDoG on two domains for which glossaries were extracted by TaxoLearn, i.e. AI and Finance. The glossaries were harvested from large collections of scholarly articles. For ProToDoG we selected 10 seeds to cover all the fields of AI, while for the financial domain we selected the same 5 seeds used in the Business

[8] Accessible from Google search with the `define:` keyword.

[9] `http://ontolearn.org` and `http://lcl.uniroma1.it/taxolearn`

Art	
rock art	includes pictographs (designs painted on stone surfaces) and petroglyphs (designs pecked or incised on stone surfaces).
impressionism	Late 19th-century French school dedicated to defining transitory visual impressions painted directly from nature, with light and color of primary importance.
point	Regarding paper, a unit of thickness equating 1/1000 inch.

Business	
hyperinflation	Extremely rapid or out of control inflation.
interbank rate	The rate of interest charged by a bank on a loan to another bank.
points	Amount of discount on a mortgage loan stated as a percentage; one point equals one percent of the face amount of the loan; a discount of one point raises the net yield on the loan by one-eighth of one percent.

Sport	
gross score	The actual number of strokes taken by a player for hole or round before the player's handicap is deducted.
obstructing	preventing the opponent from going around a player by standing in the path of movement.
points	a team statistic indicating its degree of success, calculated as follows: 2 points for a win (3 in the 1994 World Cup), 1 point for a tie, 0 points for a loss.

Table 4: An excerpt of the resulting multi-domain glossary obtained with ProToDoG.

domain of our experiments above (cf. Section 3).

We show the number of extracted terms and glosses for ProToDoG and TaxoLearn in Table 5. We also show the precision values calculated on a random sample of 5% of terms and glosses. As can be clearly seen, on both domains ProToDoG extracts a number of terms and glosses which is an order of magnitude greater than those obtained by TaxoLearn, while at the same time obtaining considerably higher precision.

6 Related Work

Current approaches to automatic glossary acquisition suffer from two main issues: i) the poor availability of large domain-specific corpora from which terms and glosses are extracted at different times; ii) the focus on individual domains. ProToDog addresses both issues by providing a joint multi-domain approach to term and glossary extraction.

Among the approaches which extract unrestricted textual definitions from open text, Fujii and Ishikawa (2000) determine the definitional nature of text fragments by using an n-gram model, 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". More recently, a domain-independent supervised approach, named Word-Class Lattices (WCLs), was presented which learns lattice-based definition classifiers applied to candidate sentences containing the input terms (Navigli and Velardi, 2010). To avoid the burden of manually creating a training dataset, definitional patterns can be extracted automatically. Faralli and Navigli (2013) utilized Wikipedia as a huge source of definitions and simple, yet effective heuristics to automatically annotate them. Reiplinger et al. (2012) experimented with two different approaches for the acquisition of lexical-syntactic patterns. The first approach bootstraps patterns from a domain corpus and then manually refines the acquired patterns. The second approach, instead, automatically acquires definitional sentences by using a more sophisticated syntactic and semantic processing. The results show high precision in both cases. However, all the above approaches need large domain corpora, the poor availability of which hampers the creation of wide-coverage glossaries for several domains. To avoid the need to use a large corpus, domain terminologies can be obtained by using Doubly-Anchored Patterns (DAPs)

	AI				Finance			
	# terms	P	# glosses	P	# terms	P	# glosses	P
ProToDoG	4983	83%	5326	84%	7370	95%	9795	96%
TaxoLearn	427	77%	834	79%	2348	86%	1064	88%

Table 5: Number and precision of terms and glosses extracted by ProToDoG and TaxoLearn in the Artificial Intelligence (AI) and Finance domains.

which, given a (term, hypernym) pair, extract from the Web sentences matching manually-defined patterns like "<hypernym> such as <term>, and *" (Kozareva and Hovy, 2010b). This term extraction process is further extended by harvesting new hypernyms using the corresponding inverse patterns (called DAP^{-1}) like "* such as <term$_1$>, and <term$_2$>". Similarly to ProToDoG, this approach drops the requirement of a domain corpus and starts from a small number of (term, hypernym) seeds. However, while DAPs have proven useful in the induction of domain taxonomies (Kozareva and Hovy, 2010b), they cannot be applied to the glossary learning task because the extracted sentences are not formal definitions. In contrast, ProToDoG performs the novel task of multi-domain glossary acquisition 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 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). ProToDoG synergistically integrates bootstrapping with probabilistic topic models so as to keep the glossary acquisition process within the target domains as much as possible.

7 Conclusions

In this paper we have presented ProToDoG, a new, minimally-supervised approach to multi-domain glossary acquisition. Starting from a small set of hypernymy seeds which identify each domain of interest, we apply a bootstrapping approach which iteratively obtains generalized patterns from Web glossaries and then applies them to the extraction of term/gloss pairs. To our knowledge, ProToDoG is the first approach to large-scale probabilistic glossary learning which jointly acquires thousands of terms and glosses for dozens of domains with minimal supervision.

At the core of ProToDoG lies our glossary bootstrapping approach, thanks to which we can drop the requirements of existing techniques such as the ready availability of domain corpora, which often do not contain enough definitions (cf. Table 5), and the manual definition of lexical patterns, which typically extract sentence snippets instead of formal glosses.

ProToDoG will be made available to the research community. Beyond the immediate usability of the output glossaries (we show an excerpt in Table 4), we also wish to show the benefit of ProToDoG in gloss-driven approaches to taxonomy learning (Navigli et al., 2011; Velardi et al., 2013) and Word Sense Disambiguation (Duan and Yates, 2010; Faralli and Navigli, 2012). The 30-domain glossaries and gold standards created for our experiments are available from `http://lcl.uniroma1.it/protodog`.

We remark that the terminologies covered with ProToDoG are not only precise, but are also one order of magnitude greater than those covered in individual online glossaries. As future work, we plan to study the ability of ProToDoG to acquire domain glossaries at different levels of specificity (i.e., domains vs. subdomains). Finally, we will adapt ProToDoG to other languages, by translating the `glossary` keyword used in step (2), along the lines of (De Benedictis et al., 2013).

Acknowledgments

erc The authors gratefully acknowledge the support of the "MultiJEDI" ERC Starting Grant 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.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

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.

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):8.

Flavio De Benedictis, Stefano Faralli, and Roberto Navigli. 2013. GlossBoot: Bootstrapping Multilingual Domain Glossaries from the Web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 528–538, Sofia, Bulgaria.

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.

Stefano Faralli and Roberto Navigli. 2013. A Java Framework for Multilingual Definition and Hypernym Extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 103–108, Sofia, Bulgaria.

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.

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 H. Hovy. 2010a. 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 and Eduard H. Hovy. 2010b. 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.

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.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international* *conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA.

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, PA, USA.

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.

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.

Mark Steyvers and Tom Griffiths, 2007. *Probabilistic Topic Models*. Lawrence Erlbaum Associates.

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):665–707.

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.