Bootstrapping a Domain-specific Terminological Taxonomy from Scientific Text

Magdalena Wolska* **Ulrich Schäfer†** **Pham The Nghia***
*Computational Linguistics Department
Saarland University
D-66041 Saarbrücken, Germany
*{magda,phamthe}@coli.uni-saarland.de
†Language Technology Lab
German Research Center for AI (DFKI)
D-66123 Saarbrücken, Germany
†ulrich.schaefer@dfki.de

Abstract

We present an approach to automated extraction of a taxonomy of domain-specific terms from scientific discourse. The approach has been developed and evaluated in the domain of computational linguistics. Concept pairs in *is-a* relation have been extracted from a subset of the ACL Anthology and WeScience. Correctness of the resource has been verified by crowdsourcing: To attract domain experts to identify correct and invalid *is-a* pairs, we used "games with a purpose". The popular games of Tetris and Invaders were modified to support concurrent and efficient annotation of domain term pairs during playing. High quality of the resulting annotations was ensured by exploiting redundancy: at least five-way agreement was required for a candidate *is-a* pair to be considered correctly extracted. Based on the crowdsourced evaluation the extraction method achieved precision around 80%.

1 Motivation and Related Work

Information on lexical relations between words or ontological relations between concepts often constitutes indispensable knowledge in language processing applications, in particular in applications involving inference. WordNet (Fellbaum, 1998) is nowadays a de facto standard source of knowledge on lexical relations between (English) words. A "Wordnet" is often one of the first resources developed for a new language.[1] Wordnets, however, are typically general-purpose lexicons with little technical terminology from specialized domains. Building domain-specific ontologies manually is a time-consuming task, requiring access to domain experts whose time is an expensive commodity.

The present work focuses on building a subset of specialized "Wordnet": a taxonomy of domain terms for computational linguistics (CL). Computational approaches to relation learning have used dictionaries (Chodorow et al., 1985), linguistic patterns (Hearst, 1992; Brin, 1999; Pantel et al., 2004; Snow et al., 2005; Yang and Callan, 2009), clustering (Caraballo, 1999; Bisson et al., 2000; Cimiano et al., 2004) and, in particular, bootstrapping as a method of pattern induction (Etzioni et al., 2005; Pantel and Pennacchiotti, 2006; Girju et al., 2006; Kozareva et al., 2008).

Building on this work, we find pairs of CL concepts in *is-a* relation with each other[2] based on bootstrapping lexico-syntactic "*is-a* patterns" from a corpus. We show that bootstrapping-based acquisition of lexico-syntactic patterns achieves satisfactory results on *OCR'ed* computational linguistics scientific papers, i.e. documents whose text is prone to have been partially corrupted. We also show that it is possible to evaluate the quality of the extracted patterns by *crowdsourcing experts*. We develop variants of two popular comuputer games which serve as "annotation tools" for evaluating the quality of the extracted patterns in a "game with a purpose" scenario.

Outline Section 2 introduces the corpora. Section 3 summarizes the taxonomy creation. Section 4 describes the games and the evalution procedures. Extraction results are outlined in Section 5. Section 6 concludes with an outlook.

2 The Corpora

Two corpora of texts in Computational Linguisitcs were used in the experiments described in this paper: a subset of the ACL Anthology[3] (8,000

[1] "Wordnets" exist or are being developed for over 50 languages; see http://www.globalwordnet.org/

[2] We will use the terms "hypernym-hyponym pairs" and "*is-a* pairs" as synonyms throughout the paper.

[3] http://www.aclweb.org/anthology

Long papers of the 9th International Conference on Terminology and Artificial Intelligence, TIA 2011, pages 17–23
Paris, 8–10 November 2011

papers) and the WeScience[4] corpus. The ACL Anthology papers were *OCR'ed* and the paper body segments were extracted. Both corpora were word- and sentence-tokenized[5] and parsed by the Stanford Parser (D. Klein and C.D. Manning, 2003). The parses and part-of-speech (POS) tags obtained from the Stanford parser were used.

3 The Approach

The approach comprises two steps: 1) domain terms are identified. 2) *is-a* patterns and pairs are bootstrapped.

3.1 Identifying Domain Terms

Multi-word domain terms were identified using the C-/NC-value method (Frantzi et al., 2000). The method proceeds in two steps: A linguistic component filters out token sequences which are unlikely to form multi-word terms using part-of-speech and lexical information. Based on experimentation the following patterns were used here: '(JJ|NOUN)+ NOUN', and '((JJ|NOUN)+ IN? (JJ|NOUN)+)* NOUN'. Then, statistical measures rank the candidates: The C-value estimates how likely a sequence is a term based on how often it occurs as a nested sub-sequence. The final ranking, NC-value, is a weighted sum of the C-value and a context factor which accounts for co-occurrence of the term with "term context words" (words likely to co-occur with domain terms).[6]

241,806 unique multi-word terms were extracted from the ACL Anthology, of which top 200K were preserved. 5,000 head nouns of these were added as *single-word domain terms*.[7]

3.2 Bootstrapping *is-a* Patterns

In general, bootstrapping approaches work by finding new patterns based on an incrementally extended set of patterns with "anchors", that is, designated pattern elements. Typically, the process starts with a small set of seed patterns and new patterns are found by fixing the anchors or the pattern bodies to specific values, in alternating cycles.

In present experiments we used anchored patterns similar to those proposed in (Pasca, 2004; Etzioni et al., 2005). The process starts with a single pattern 'TERM such/JJ as/IN TERM' which is known to be a reliable indicator of the *is-a* relation. First, for each pattern instance, p, in the set of patterns, a set of pairs which it extracts, ISA_p, is found by matching the pattern against the corpus.[8] The extracted pairs are ranked and the top-N are added to the ISA result set. Then, the top-N pairs are used to instantiate the anchors in order to find new pattern bodies. These are also ranked. Finally, the top new pattern is added to the pattern set and the process is repeated with the extended pattern set. The pseudo-code of the bootstrapping procedure is included below:

Notation:
  C: the corpus (see Section 2)
  TERM: a domain term (see Section 3.1)
  P, NewPatterns: sets of patterns
  * (asterisk): wildcard for the pattern's body
  ISA, ISA_p : sets of *isa* pairs

begin
  ISA = ∅, P = {'TERM such/JJ as/IN TERM'}
  repeat
    foreach p in P
      Find ISA_p={ <TERM, TERM, p> | p matches in C}
      Rank all <x, y, p> in ISA_p (see Section 3.3)
      Add top-N instances to ISA
      Find NewPatterns={'x * y' | top-N match in C}
      Rank all np in NewPatterns (see Section 3.3)
      Add top-ranked np in NewPatterns to P
  until No. iterations == i
  return {ISA, P}
  foreach isa in ISA
    Mark isa-direction (see Section 3.4)
end

3.3 Ranking

The discovered *is-a* instances and patterns are ranked using two measures: *reliability* of instances and patterns and pattern *productivity*.

Productivity of a pattern p, $p(p)$, is defined as the proportion of *is-a* pairs extracted by the pattern, $|ISA_p|$, out of all pairs which can be extracted:

$$p(p) = \frac{|ISA_p|}{|ISA|}$$

Reliability of a pattern p, $r_p(p)$ accounts for the

[4] http://wiki.delph-in.net/moin/WeScience

[5] Existing sentence segmentation of WeScience was used. No evaluation of the OCR quality was performed.

[6] Please refer to (Frantzi et al., 2000) for details.

[7] Head nouns were identified using syntactic parses of multi-word terms. At this stage, the domain term identification method has not been rigorously evaluated.

[8] The pattern which extracted the given pair is stored together with the pair.

strength of association between the concepts in an *is-a* pair which a pattern extracts, and is defined, following (Pantel and Pennacchiotti, 2006), as:

$$r_p(p) = \frac{\sum_{i \in I} (\frac{pmi(i,p)}{max_{pmi}} * r_i(i))}{|I|}$$

where $pmi(i,p)$ is the pointwise mutual information between an *is-a* pair instance and a pattern p,[9] max_{pmi} is the maximum pmi between all patterns and all *is-a* instances, $|I|$ is the number of instances, and $r_i(i)$ is the reliability of an instance (see below). The final rank of a pattern is a sum of its productivity and reliability.

Reliability of an is-a instance i, $r_i(i)$, is defined analogously to $r_p(p)$:

$$r_i(i) = \frac{\sum_{p \in P} (\frac{pmi(i,p)}{max_{pmi}} * r_p(p))}{|P|}$$

$|P|$ is the number of patterns.

The following illustrate the extracted patterns:[10]

(TERM | coord. TERMS) and other TERM

TERM (e.g.|for instance|for example) (TERM|CC-TERMS)

TERM (especially|including|like|such as) (TERM|CC-TERMS)

such TERM as (TERM | CC-TERMS)

TERM is a TERM

3.4 Identifying Hypernyms

The resulting pairs are processed using pattern-specific rules in order to identify the hypernym and the hyponym. The rules match the patterns' forms and were created by hand by inspecting a subset of pattern instances and *ordered* with decreasing specificity. Examples of rules for the 'such as' patterns are shown below:[11]

---

1. 'TERM_1 such as TERM_2' → TERM_2 *is-a* TERM_1

2. 'NP_1 IN NP_2 such as NP_3' → NP_3 *is-a* NP modified by an adjective expressing quantity (e.g. "many", "numerous") or difference (e.g. "different", "other", "various")

3. 'NP_1 IN NP_2 such as NP_3' & head(NP_1) is "kind", "type", "number", "variety" → NP_3 *is-a* NP_2

4. 'NP_1 * S * such as NP_3' → NP_1 *is-a* NP_2

5. Of all the NPs preceding 'such as' the one with highest pmi value with the NP following 'such as' is the hypernym

---

Note that the rules refer to the Stanford Parser's output. In order to make this possible we store the parsed sentences from which a pattern was extracted together with the pattern.

In total, 9,565 candidate *is-a* pairs have been extracted. Examples of extracted pairs (both valid and invalid) are shown in Table 1.

3.5 Building the Taxonomy

A simple taxonomy is created from the set of extracted *is-a* pairs by iterating over the list of candidates ordered by their reliability as follows: Starting with an empty taxonomy, 1) an ISA pair from the reliability-ordered list is temporarily added to the taxonomy, 2) the taxonomy is tested for cycles, 3) if a cycle is found, the temporarily added pair is removed, otherwise it is preserved in the taxonomy, 4) the process proceeds with the next pair.

4 Evaluation Method

Precision of the *is-a* pair extraction was evaluated in the "games with a purpose" (GWAP) paradigm (von Ahn and Dabbish, 2008) by crowdsourcing domain experts. Two popular games, Tetris and Invaders, were modified to support concurrent and efficient annotation of the extracted domain term pairs during playing.[12]

4.1 The Games

Tetris The standard Tetris setup was modified in that the falling bricks were labelled with domain terms (hyponym candidates) and the wall area was divided into two parts by a fence with each side labelled with two other domain terms (hypernym candidates). Each combination of a brick label

[9] $pmi(i,p) = log \frac{|x,p,y|}{|x,*,y||*,p,*|}$. $|x,p,y|$ is the frequency of pattern p instantiated with pair <x,y>.

[10] Patterns are simplified for presentation. CC-TERMS denotes a noun phrase group (in the parser output) formed using coordination: coordinating conjunction and/or commas.

[11] The rules have been simplified for presentation. NP denotes a noun phrase, IN a preposition, head(NP) the head of a noun phrase, and S the clause tag. Wildcard matching is non-greedy.

[12] The games were based on open source implementations available at `http://code.google.com/p/java-tetris/` (Tetris) and `http://sourceforge.net/projects/matharcade/` (Invaders)

Hypernym	Hyponyms
natural language processing application	information extraction, question answering, machine translation, information retrieval, document summarization, speech recognition, pos tagging, named entity recognition, question answering system, open-domain question-answering, text mining, named entity extraction, question-answering, automatic lexical acquisition, text summarization, document clustering, language model building, word sense disambiguation, annotation projection, cross language information retrieval, ...
agglutinative language	korean, basque, chinese, hungarian, japanese, thai
web search engine	google, yahoo, altavista
classifier	svm, decision tree, support vector machine, naive bayes, conditional random field, maximum entropy classifier, dependency path, probabilistic classifier, pruned decision tree, timbl, k-nn, acoustic confidence score
vector distance measure	euclidean distance, cosine
dependency relation	subj, subject, object, arg, obj, head-modifier
open-class word	adjective, adverb, verb, common noun, proper name
morphological feature	number, gender, person, case, aspect, pos, tense, count, voice
sequence labeling task	named entity recognition, pos tagging, chunking, syntactic chunking
evaluation metric	nist, bleu

Table 1: Examples of extracted hypernym-hyponym pairs (including invalid pairs)

and a wall label was an extracted candidate *is-a* pair. Tetris was designed as a positive selection game: the task was to indicate terms which *were* hyponyms of one of the given hypernym candidates by placing the falling brick on the appropriate side of the fence. All the standard Tetris rules applied. If a brick did not match any of the walls (i.e. the two extracted pairs were possibly invalid) the players were asked to indicate this by pressing the space key. A screen-shot of the Tetris game is shown in Figure 1 on the left.

Invaders Similarly, the Invaders setup was modified in that the ships were labelled with domain terms (hyponym candidates) and the specific hypernym was displayed next to the cannon. The Invaders setup lends itself obviously to negative selection: the task was to shoot ships labeled with terms which *were not* hyponyms of the given hypernym candidate. A screen-shot of the Invaders game is shown in Figure 1 on the right.

In the course of a game, the target hypernym candidates (the wall labels and the cannon label) were changed in order to avoid monotony of game play and to ensure broader coverage of the evaluation. Both games were set up with a predefined playing time, selected by the player.

Aside from the two GWAPs we also provided a **True-or-False Quiz** for those participants who were willing to contribute their time and expert knowledge, but were not interested in the games.

4.2 Participants and Scoring

The evaluation was advertised as a competition with prizes. The participants were recruited on voluntary basis from colleagues and students at the university's Computational Linguistics department and from a Language Technology department of an on-campus research institute. All the participants were required to register with the games by choosing a unique player ID and to fill out a questionnaire on their computational linguistics/language processing background.

The scores were calculated as follows: A small set of pairs were annotated by the authors of the paper before the competition in order to obtain an initial annotated subset (single annotation per item). When no information about the pair was known, one point was given to the player for every annotated pair to encourage playing. As more players participated and pairs were annotated by more than one player the score given for the pair

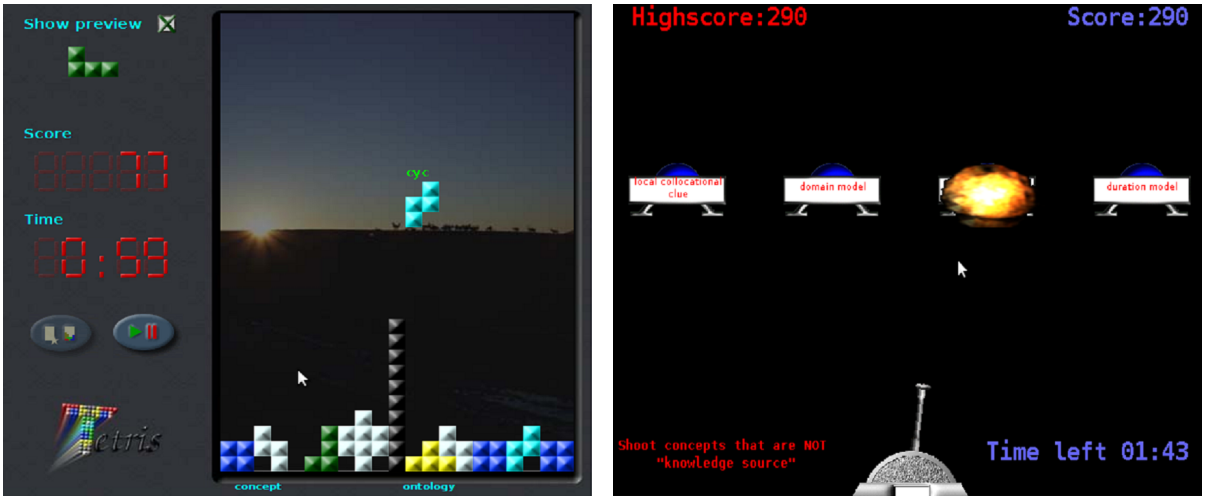

Figure 1: Screen-shots of the modified games: Tetris (left) and Invaders (right)

was set to the number of agreements; e.g. if one player previously annotated the same way as the current player (i.e. two players agreed) two points were given to the current player. In case of disagreements on a pair, the score was set to the greater of the number of positive or negative agreements.[13] Each pair was pooled for annotation by different players until a pre-defined agreement threshold was reached; see below.

4.3 Efficiency and Quality Control

With just under 10,000 candidate pairs to be verified, the annotation task becomes time-consuming and random sampling from a set of this size inefficient. In order to ensure efficiency of the crowdsourced evaluation, an item pooling procedure was introduced. At any given time during the competition a fixed-size subset of the data set was used as the target pool for verification. Items were removed from the target pool once a pre-defined number of agreements was reached. The agreement threshold served as a quality control measure: While game play may be an entertaining alternative to dry annotation, it may also introduce errors; be it due to the players' greater focus on the entertainment aspect rather than annotation, the difficulty in making a decision in problematic cases, or simply due to the time pressure in the limited play time. In the present experiment the minimum agreement threshold was set at five; i.e. five players/annotators must agree that a pair is valid/invalid ("5-way agreement").

4.3.1 Sampling Evaluation Instances

At each time, the set of extracted candidate pairs (DATA) is divided into four subsets: a fixed-size target pool (TARGET-POOL), a fixed-size reserve (RESERVE-POOL), done pool (DONE-POOL), and the remaining unannotated set (UNANNO-POOL). TARGET-POOL is the set of items which need to be verified; i.e. 5-way agreement not reached. RESERVE-POOL contains also unverified data and is used in case a given player has already annotated all the items from the current TARGET-POOL, but is still playing (i.e. needs more data). Once an agreement threshold has been reached on an item in the TARGET-POOL or RESERVE-POOL, it is moved to the DONE-POOL. Data sampled for a user at the start of his/her game, GAME-DATA, is randomy selected from, first, TARGET-POOL, then, if TARGET-POOL is empty, RESERVE-POOL, then from DONE-POOL.

In short, user data for all users is sampled from TARGET-POOL, RESERVE-POOL, and DONE-POOL, in this order, in small subsets until all DATA is in DONE-POOL. Items are presented to a user at most once. The sampling subset size was set at 50 instances. The sampling and item status update procedure is summarized as pseudo-code below:[14]

[13] By "positive" and "negative" agreement we mean that at least two players considered a candidate pair valid or invalid, respectively.

[14] "user game" is a game session of a duration specified by the user. Sampling from DONE-POOL is omitted.

GAME-DATA: dynamically updated user game data
DATA = TARGET-POOL ∪ RESERVE-POOL ∪ DONE-POOL ∪ UNANNO-POOL

begin
  case
    start game //first sample for this user game
      if TARGET-POOL $\neq \emptyset$
        Select up to 50 randomly from TARGET-POOL as GAME-DATA
      else
        Select up to 50 randomly from RESERVE-POOL as GAME-DATA
    game running //user annotated all the GAME-DATA
      Select up to 50 randomly from RESERVE-POOL
    game finished //update agreements for user data
      foreach isa in GAME-DATA
        if number of agreements(isa) ≥ 5
          Move isa to DONE-POOL
          Move random item from UNANNO-POOL to TARGET-POOL or RESERVE-POOL[a]
end

[a]Depending on where "done" isa stemmed from.

4.3.2 Results Verification by Players

As an additional quality control measure explicit results verification by users was introduced. After a finished game, a player was redirected to a page with all of his/her choices made during game play. Each item's annotations were presented as radio-buttons which the user could modify (e.g. change a pair's status from valid to invalid). Results verification was optional; a "Skip verification" button took the user back to the games' front page. Because, depending on the game duration, the list may have been long, verification was not always performed. Less then half of the annotated data set was explicitly verified by players this way.

5 Results

The competition for prizes lasted 10 days. During this time, 61 players participated. Tetris was more popular than Invaders (32 players vs. 10). The True-or-false quiz was selected by 26 participants. Only one participant played all the games.

The summary of evaluation data and the results are shown in Table 2. Almost a third of the entire data set was presented to the participants within 10 days (2,940 out of the 9,565 pairs). Almost 7,000 (redundant) annotations were obtained; on average two annotations per item.

The table shows precision results for 3-way and 5-way agreements: three or five, respectively, of the same annotations by different players of an *is-a* pair; be it positive agreements (valid *is-a*) or negative (invalid *is-a*). The obtained precision was 77% for 3-way and 80% for 5-way agreements.

	Category	Value
Data statistics	No. presented pairs	2940
	% of entire set	31%
	No. annotations	6782
Precision results	No. 3-way agreements	639
	of which, no. valid *is-a* pairs	490
	no. invalid pairs	149
	3-way precision	**77%**
	No. 5-way agreements	298
	of which, no. valid *is-a* pairs	239
	no. invalid pairs	59
	5-way precision	**80%**

Table 2: Summary of the evaluation data and the results

Discussion Considering the unsupervised nature of the bootstrapping procedure, we believe the result in the high 80s is satisfactory. We would have preferred more annotations to have been explicitly verified by the participants using the procedure presented in Section 4.3.2, so that we can be even more certain of the quality of the 5-way agreements. However, explicit verification of this kind conflicts, of course, with the entertainment aspect of the evaluation procedure.

Now, almost a third of the entire data set could have been evaluated using the GWAP method within 10 days. This result appears encouraging to consider exploiting the method further in a more broadly advertized competition. At this point, however, we did not continue the present evaluation due to certain problems (summarized below) which we would like to solve first.

Known Problems First, we noticed that certain too general concepts were included as members of *is-a* pairs. Examples of these are: "thing", "something", "information", or "concept". Arguably "thing" and "concept" may be considered domain terms in the area of ontologies. The other two are incorrectly identified domain terms. Therefore, we would like to evaluate the term identification procedure more rigorously and improve it. In particular, identification of the single-word terms as it is currently done (including all heads by default) is sub-optimal. Although some invalid terms are filtered out by the *is-a* ranking procedure (Section 3.3), clearly improvement of the initial term

identification would lead to better results.

Second, after the competition some players commented on the dark graphics and the hard to read brick and space-ship labels (the terms themselves). This is a simple technical issue which can be easily solved before a follow-up evaluation.

6 SUMMARY AND FURTHER WORK

We presented an unsupervised method of extracting pairs of domain terms in *hypernym-hyponym* relation (concepts in *is-a* relation) from scientific papers in a restricted domain (computational linguistics/natural language processing). The method is based on an iterative bootstrapping process whereby anchored patterns are acquired from text and ranked according to their reliability. The pairs are used to build up a domain-specific taxonomy of terms. The precision of the extraction method was evaluated in a "games with a purpose" scenario by crowdsourcing expert knowledge and achieved precision of around 80%. The extraction method itself is generic and can be applied to other domains than computational linguistics.

We are planning a follow-up larger-scale evaluation in the same scenario. First, we plan to use the data we already have in order to improve the bootstrapped patterns. We will manually verify the 5-way agreement pairs in order to obtain a reasonably-sized set of valid *is-a* pairs. Second, the valid pairs will be used to bootstrap new *is-a* patterns in a new round of bootstrapping experiments exploring also "doubly-anchored" patterns (Kozareva et al., 2008) and using an improved ranking scheme. We are planning to make the resulting data available to the community as part of the LT World[15] knowledge portal.

Acknowledgments

The authors thank the participants of the online games. Ulrich Schäfer's work has been funded by the German Federal Ministry of Education and Research (contract 01IW08003, project TAKE).

References

G. Bisson, C. Nédellec, and D. Cañamero. 2000. Designing clustering methods for ontology building: The Mo'K Workbench. In *Proc. of the ECAI Ontology Learning Workshop*, pages 13–19.

S. Brin. 1999. Extracting Patterns and Relations from the World Wide Web. In *Proc. of the Workshop on the World Wide Web and Databases*, pages 172–183.

S. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proc. of ACL*, pages 120–126.

M. Chodorow, R. Byrd, and G. Heidorn. 1985. Extracting semantic hierarchies from a large on-line dictionary. In *Proc. of ACL*, pages 299–304.

P. Cimiano, A. Hotho, and S. Staab. 2004. Comparing Conceptual, Divisive and Agglomerative Clustering for learning Taxonomies from Text. In *Proc. of the 16th ECAI Conference*, pages 435–439.

D. Klein and C.D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proc. of ACL*, pages 423–430.

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

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

K. Frantzi, S. Ananiadou, and H. Mima. 2000. Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, 3:115–130.

R. Girju, A. Badulescu, and D. Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proc. of the 14th Coling Conference*, pages 539–545.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proc. of ACL*, pages 1048–1056.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proc. of the 21st Coling and the 44th ACL Meeting*, pages 113–120.

P. Pantel, D. Ravichandran, and E. Hovy. 2004. Towards terascale knowledge acquisition. In *Proc. of the 20th Coling Conference*, pages 771–777.

M. Pasca. 2004. Acquisition of categorized named entities for web search. In *Proc. of the 13th CIKM Conference*, pages 137–145.

R. Snow, D. Jurafsky, and A. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*, pages 1297–1304.

L. von Ahn and L. Dabbish. 2008. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67.

H. Yang and J. Callan. 2009. A metric-based framework for automatic taxonomy induction. In *Proc. of the 47th ACL Meeting and the 4th the AFNLP Conference*, pages 271–279.

[15] http://www.lt-world.org