# A graph-based method to improve WordNet Domains

Aitor González, German Rigau
IXA group UPV/EHU, Donostia, Spain
agonzalez278@ikasle.ehu.com
german.rigau@ehu.com

Mauro Castillo
UTEM, Santiago de Chile, Chile
mcast@informatica.utem.cl

**Abstract.** WordNet Domains (WND) is a lexical resource where synsets have been semi-automatically annotated with one or more domain labels from a set of 170 hierarchically organized domains. The uses of WND include the power to reduce the polysemy degree of the words, grouping those senses that belong to the same domain. This paper presents a novel automatic method to propagate domain information through WordNet. We compare both labellings (the original and the new one) allowing us to detect anomalies in the original WND labels. We also compare the quality of both resources (the original labelling and the new one) in a common Word Sense Disambiguation task. The results show that the new labelling clearly outperform the original one by a large margin.

## 1 Introduction

Building large and rich knowledge bases is a very costly effort which involves large research groups for long periods of development. For instance, hundreds of person-years have been invested in the development of wordnets for various languages [1].

WordNet Domains[1] (WND) is a lexical resource where synsets have been semi-automatically annotated with one or more domain labels from a set of 165 hierarchically organized domains [2, 3]. WND allows to reduce the polysemy degree of the words, grouping those senses that belong to the same domain [4].

But the semi-automatic method used to develop this resource was not free of errors and inconsistencies. For instance, noun synset <diver$_n^1$ frogman$_n^1$ underwater_diver$_n^1$> defined as *someone who works underwater* has domain *history* because it inherits from its hypernym <explorer$_n^1$ adventurer$_n^2$>. WND has never been verified manually. Additionaly, WND is aligned to WordNet 1.6[5], and there is no version for 3.0.

[1] http://wndomains.fbk.eu/

We suggest a novel graph-based approach for improving WND. As a result we obtained a new semantic resource derived from WordNet Domains and aligned to WordNet 3.0.

After this short introduction, Section 2 describes a very simple method of inheritance used to fill the gaps that have arisen due to the porting process from WordNet 1.6 to 3.0. In section 3 we describe our novel graph-based method, based on the UKB algorithm, used to generate new domain labels aligned to WordNet 3.0. Section 4 presents an example of how to evaluate in a semi-automatic way the quality of the domain labels assigned in the original WND. Finally, section 5 presents an evaluation of the new domain labelling based on a common Word Sense Disambiguation task.

## 2 Domain inheritance

WND was developed using WordNet 1.6. One consequence of the automatic mapping that we used to upgrade version 1.6 to 3.0 is that many synsets were left unlabeled (because there are new synsets, changes in the structure, etc.).

Thus, the first tasks undertaken has been to fill these gaps. For them, we have carried out a propagation of the labels by inheritance of nominal and verbal synsets. In WordNet, the adjectives are organized in terms of binary oppositions (antonymy) and similarity of meaning (synonymy). The structure of WordNet for adjectives and adverbs makes this spread not trivial. Therefore this simple process has been not carried out neither for adjectives nor for adverbs.

Consider the example shown in Figure 1. For nouns and verbs, we have worked on the assumption that synsets are mostly correctly labeled, and therefore we have worked exclusively on those synsets that had no labels at all. We inherited the label or labels from its hypernyms. If a synset has more than one hypernym, the domain labels are taken from all of them. During this phase has been taken into account the incompatibility between domain labels, preventing the same synset can be, for instance, both *factotum* and *biology*.

This process increased our domain information by nearly a 18-19%, as shown in Tables 1 and 2:

**Table 1.** Number of synsets with domain labels.

| PoS | Before | After | Increase |
|---|---|---|---|
| Nouns | 66,595 | 83,286 | +25% |
| Verbs | 12,219 | 14,224 | +16% |
| All | 100,315 | 119,011 | +19% |

However, this process may also have propagated innapropriate domain labels to unlabeled synsets. In the next section we present some examples using a

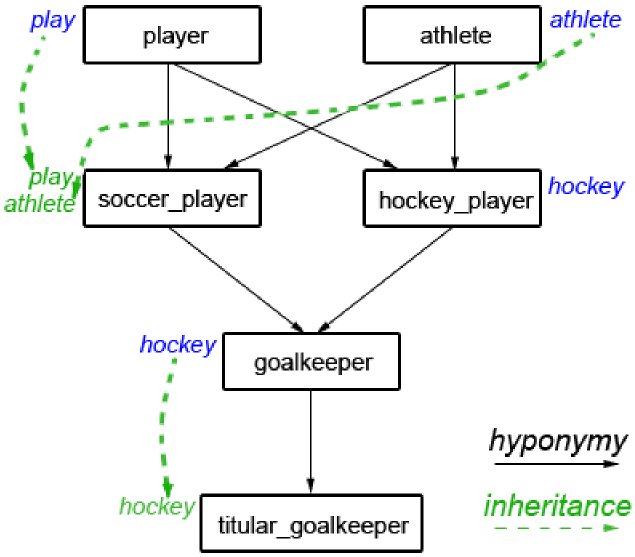


**Fig. 1.** Example of inheritance of domain labels.

**Table 2.** Total number of domain labels.

| PoS | Before | After | Increase |
|---|---|---|---|
| Nouns | 87,938 | 108,665 | +24% |
| Verbs | 13,026 | 15,051 | +16% |
| All | 124,551 | 146,899 | +18% |

new graph-based method for propagating domain labels through WordNet. Additionaly, the method can also be used to detect anomalies in the original WND labels.

## 3 A new graph based method

UKB[2] algorithm [6] applies personalized PageRank on a graph derived from a wordnet. This algorithm has proven to be very competitive on Word Sense Disambiguation tasks and it is easily portable to other languages that have a wordnet [7]. Now, we present a novel use of the UKB algorithm for propagating information through a wordnet structure.

Given an input context, '*ukb_ppv*' (*Personalized PageRank Vector*) algorithm outputs a ranking vector over the nodes of a graph, after applying a *Personalized*

[2] http://ixa2.si.ehu.es/ukb/

*PageRank* over it. We just need to use a wordnet as a knowledge base and pass to the application the contexts we want to process, performing a kind of *spreading activation* through the structure of a wordnet.

As a context we used those synsets labelled with a particular domain. Thus, for each of the 169[3] domain labels included in the MCR we generated a context. Each file contains the list of offsets corresponding to those synsets with a particular domain label. After creating the context file, we just need to execute '*ukb_ppv*' that will return a ranking of the weights for each wordnet synset with respect to that particular domain.

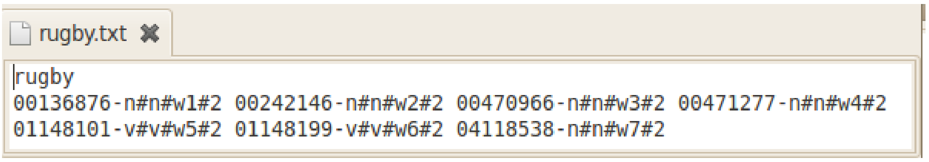
Once made the process for all domains we will have the weight of each synset for each of the domains. Therefore, we know which are the highest weights for each domain and the highest weights for each synset. This allows us to estimate which synsets are more representative of each domain (those who have more weight in the ranking) and which domains are best for each synset (those who have attained a higher weight for that synset).

Basically, what we do is to mark some synsets with a domain (using the labels we already know from the original porting process) and use the wordnet graph to propagate the new labelling. We work on the assumption that a synset directly related to several synsets labelled with a particular domain (i.e *biology*) would itself possibly be also related somehow to that domain (i.e. *biology*). Therefore, it makes no sense to use the domain *factotum* for this technique.

### 3.1 Propagating domain labels

We have generated two different knowledge bases. The first one only contains the original WordNet relations. The second one, also contains the relationships between glosses, increasing the size and richness of the knowledge base. Instructions for preparing the binary databases for UKB using WordNet relations are inside the downloadable file[4] of the UKB package.

It has been necessary to generate a context file for each domain. Generating a context is as simple as creating a text file with the synset offsets that have the domain label. An example of a context file for the *rugby* domain can be seen in Figure 2. We can see a list of offsets representing synset of the Table 3.



```
rugby
00136876-n#n#w1#2 00242146-n#n#w2#2 00470966-n#n#w3#2 00471277-n#n#w4#2
01148101-v#v#w5#2 01148199-v#v#w6#2 04118538-n#n#w7#2
```

**Fig. 2.** View of the format of a context file.

[3] Excluding *factotum* labels.

[4] http://ixa2.si.ehu.es/ukb/

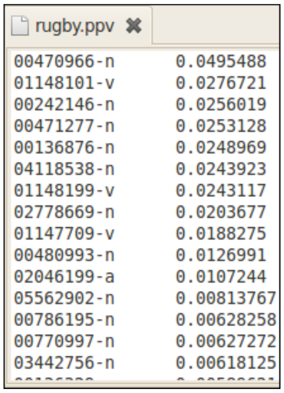
**Table 3.** List of synset with ”*rugby*” as domain label.

| Synset | Variants |
|---|---|
| eng-30-00136876-n | goal-kick |
| eng-30-00242146-n | scrum, scrummage |
| eng-30-00470966-n | rugby, rugby_football, rugger |
| eng-30-00471277-n | knock_on |
| eng-30-01148101-v | hack |
| eng-30-01148199-v | hack |
| eng-30-04118538-n | rugby_ball |

One of the problems that comes up when analyzing the results is that the own domain labels of a synset have an unbalanced weight on the final ranking of that synset. Almost always the own labels of a synset appear in the top positions. In order to avoid this undesired effect, we generated new contexts, specific to each synset, and each domain. Thus, a synset can not vote for its own domains and only the rest of synsets decide the final weights of the ranking.

### 3.2 Post-processing

Once generated the context files, the UKB algorithm is executed. The result is a list with the weight for each synset for a domain. The next step is to sort the file by weight, highlighting those synsets that are more representative of the domain (Figure 3).



```
00470966-n      0.0495488
01148101-v      0.0276721
00242146-n      0.0256019
00471277-n      0.0253128
00136876-n      0.0248969
04118538-n      0.0243923
01148199-v      0.0243117
02778669-n      0.0203677
01147709-v      0.0188275
00480993-n      0.0126991
02046199-a      0.0107244
05562902-n      0.00813767
00786195-n      0.00628258
00770997-n      0.00627272
03442756-n      0.00618125
```

**Fig. 3.** Result of a PPV ranking sorted by weight (only the first lines are shown).

Furthermore, we can sort the result by synset. This allows us to, once we have a file for each domain, put them together in a matrix. Each line of this matrix will represent a synset, and the columns will be weights corresponding to each domain. The highest values of a line (synset) will be the more representative domains for that synset.

Table 4 shows the first ten domains and weights resulting from the application of this method on synset <diver$_n^1$ frogman$_n^1$ underwater_diver$_n^1$> originally labeled as *hystory*, which seem to be incorrect. The suggestions of the algorithm seems to improve the current labeling because it suggests *sub* (possibly the best one) and *diving* (possibly, the second best option). Moreover, the method suggests the wrong label with a much lower weight.

**Table 4.** PPV weight rankings for sense *diver*$_n^1$.

| **Weight** | **Domain** |
|---|---|
| 0.0144335: | sub |
| 0.0015939: | diving |
| 0.0001725: | swimming |
| 0.0001297: | history |
| 0.0000557: | nautical |
| 0.0000529: | fashion |
| 0.0000412: | jewellery |
| 0.0000315: | ethnology |
| 0.0000274: | archaeology |
| 0.0000204: | gas |

## 4 Analyzing ranking changes

It seems that the algorithm is able to generate a ranking in which the most appropriate labels obtain larger weights and also that avoiding the own labels of a synset reduces the weights for incorrect domain labels.

In the next experiment we study how to evaluate in a semi-automatic way the quality of the original labelling. To do that we check the domain labels of the synsets, taking into account the position they occupy in the weight vector. If a synset has '*n*' domain labels, the displacement is calculated for every label. For example, if a synset has two labels and one of the domains occupies the first position and the other the third one, they receive an offset of +0 and +1 respectively. That is, we calculate how many positions they moved from its original place. All those labels with an offset of six or greater are considered in the same group. Possibly, this test will allow us to discover wrong labeled synsets (or at least delimit the search) or to create a group of labels with a high value of reliability.

Therefore we tested the process for each PoS. The results obtained are in the Table 5.

**Table 5.** Method WN+gloss: Displacement of domain labels regarding their current position (separated by PoS).

| | Offset | | | | | | |
|---|---|---|---|---|---|---|---|
| **PoS** | **0** | **1** | **2** | **3** | **4** | **5** | **6+** |
| **Nouns** | 55.52% | 18.51% | 10.06% | 5.19% | 1.95% | 1.95% | 6.82% |
| **Verbs** | 40.46% | 15.95% | 13.39% | 7.69% | 4.56% | 0.85% | 17.09% |
| **Adjectives** | 51.04% | 21.35% | 8.85% | 2.60% | 1.56% | 4.17% | 10.42% |
| **Adverbs** | 60.40% | 13.86% | 5.94% | 0.99% | 4.95% | 2.97% | 10.89% |
| **Total** | **54.48%** | **18.60%** | **10.07%** | **5.04%** | **2.06%** | **2.10%** | **7.65%** |

Detecting the labels that have been displaced six or more positions (Table 5) allows us to recognize possible synset that have been labeled incorrectly. An example can be seen in Table 6.

Results for 'ili-30-00747215-n':

- **Variants:** pornography_1 porno_1 porn_1 erotica_1 smut_5
- **Gloss:** creative activity (writing or pictures or films etc.) of no literary or artistic value other than to stimulate sexual desire
- **Domains:** law

**Table 6.** Method WN+gloss: UKB weight rankings for sense 1 of "*porno*".

| Method WN+gloss | |
|---|---|
| **Weight** | **Domain** |
| 0.000123453: | sexuality |
| 0.000112444: | cinema |
| 0.000077780: | theatre |
| 0.000075525: | painting |
| 0.000062377: | telecommunication |
| 0.000060640: | publishing |
| 0.000050370: | psychological_features |
| 0.000047003: | photography |
| 0.000046853: | artisanship |
| 0.000040458: | graphic_arts |

The example in Table 6 shows how the label *law* (incorrectly assigned) disappears from the first ten positions of the list. Instead, the algorithm suggests *sexuality* and *cinema*, which in this case seems to be much more appropriate.

## 5 Evaluation

To evaluate the new resources, we decided to compare the original labelling against the new domain labels that we have generated in a common Word Sense Disambiguation task.

Senseval-3 task 12 *Word-Sense Disambiguation of WordNet Glosses*[5] was designed as an all-words task using as a gold standard the handtagged words provided by the eXtended WordNet [8].

Similarly, we selected as a gold standard, a random subset of 933 disambiguated words from the semantically disambiguated WordNet glosses[6]. This sample is available at http://adimen.si.ehu.es/web/XWND.The task is to try to select the correct sense of a target word appearing in its gloss. For example, consider synset <tortoiseshell$_n^3$ tortoiseshell-cat$_n^1$ calico_cat$_n^1$> defined as a *cat having black and cream-colored and yellowish markings*. In this case, we should try to disambiguate which of the seven senses of the word *cat* is the one used in the gloss.

Our approach follows heuristic 5 from [9]. Having a synset with a particular WordNet Domain label, this method selects those synsets from the target word of the gloss having the same Domain label. According to [9] this heuristic obtained a precision of 69.7%, a recall of 18.9% and it was applied only 27.1% of the cases on the WordNet 2.0 dataset.

The technique consists in choosing the synset that shares the domain labels with the synset defined by the gloss we are trying to disambiguate. At this point we must differentiate between the original labelling and those generated using our graph technique. One advantage of our labelling is that we have a ranking of the 169 domain labels, while the old labelling provides a limited number of labels. In the case of the original labels of WordNet Domains (WND) all available domain labels are used for the disambiguation task (varying between one and four domain labels per synset). In the case of the new labeling, we will check the results obtained using between one and five labels for the disambiguation task. We also use the *scorer2* software available on the Senseval-3 website[7].

For those cases of multiple matches in the candidate synsets (when more than one synset shares the domain labels) we will choose those sharing more labels (where possible). That is, if we are using three domain labels to disambiguate, we will select as candidates those synsets that share the three labels with the synset defined by the gloss we are trying to disambiguate. In the case of ties, we choose all those synsets that matches (which decreases the score obtained after applying the software *scorer2*). If any of the candidate synsets shares the three domain labels, we will select all those who share two labels, and so on.

Following the example of the synset <tortoiseshell$_n^3$ tortoiseshell-cat$_n^1$ calico_cat$_n^1$>, labeled with *animals* and *biology* domain labels, the chosen senses will be synsets <cat$_n^1$ true_cat$_n^1$> and <cat$_n^7$ big_cat$_n^1$>. The two chosen synsets

[5] http://www.clres.com/SensWNDisamb.html

[6] http://wordnet.princeton.edu/glosstag.shtml

[7] http://www.senseval.org/senseval3

share two domain labels with the synset <tortoiseshell$^{3}_{n}$ tortoiseshell-cat$^{1}_{n}$ calico_cat$^{1}_{n}$> (*animals* and *biology*). None of the other senses of *cat* shares any of the domain labels with the synset <tortoiseshell$^{3}_{n}$ tortoiseshell-cat$^{1}_{n}$ calico_cat$^{1}_{n}$>.

After performing this operation with the 933 glosses we will get the values for precision, recall and F1 score (Table 7). Nomenclature employed is as follows:

- **Method 0**: Disambiguation performed using the original WordNet Domains.
- **Method 1**: Disambiguation performed using the new labelling obtained using UKB and WordNet relations as a knowledge base (WN).
- **Method 2**: Disambiguation performed using the new labelling obtained using UKB and WordNet relations enriched with relations between glosses (WN+gloss).

**Table 7.** Precision, recall and F1 values obtained using *scorer2*.

| | **Method 0** | | | **Method 1** | | | **Method 2** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Label #** | **P** | **R** | **F1** | **P** | **R** | **F1** | **P** | **R** | **F1** |
| 1 | - | - | - | 0.779 | 0.242 | 0.369 | 0.796 | 0.283 | 0.418 |
| 2 | - | - | - | 0.739 | 0.373 | 0.496 | 0.795 | 0.509 | 0.621 |
| 3 | - | - | - | 0.720 | 0.435 | 0.542 | 0.807 | 0.654 | 0.722 |
| 4 | - | - | - | 0.695 | 0.474 | 0.564 | 0.793 | 0.693 | 0.740 |
| 5 | - | - | - | 0.682 | 0.504 | 0.580 | 0.796 | 0.745 | 0.770 |
| All | 0.668 | 0.319 | 0.432 | - | - | - | - | - | - |

Additionally, if we look at the plots with all the values obtained for the new domains we will see that both methods outperform the original WordNet Domains (the line for the *Method 0* is shown as baseline). The plots are shown in Figure 4 (precision), 5 (recall) and 6 (F1 score). *Method 2* seems to be the most robust of the three, reaching an F1 score of 0.770 when using three domain labels.

## 6 Concluding Remarks

We have presented a new robust graph-based method which propagates domain information through WordNet. Firtly, we described a simple inheritance mechanism to complete unlabelled synsets from WordNet 3.0. Secondly, we provide some examples of the new domain labellings focussing on those synsets which provided larger variations. Thirdly, an empirical evaluation has been carried out in a common Word Sense Disambiguation task. On this task, the heuristic using the new WordNet Domains clearly outperforms by a large margin the one using the original WordNet Domains.

After these initial empirical tests, we drawn some preliminary conclusions:

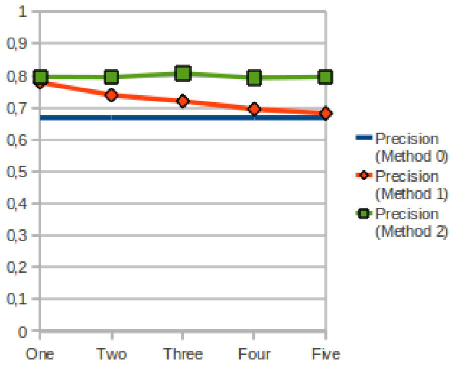


**Fig. 4.** Graphic showing *precision* values.

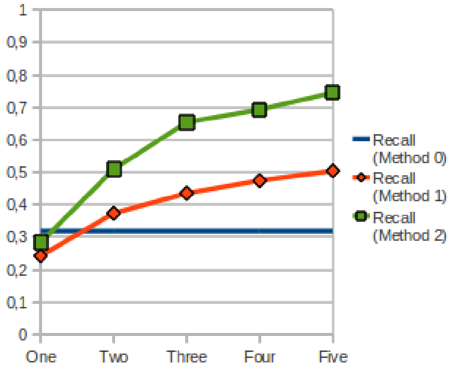


**Fig. 5.** Graphic showing *recall* values.

1. The propagation method seems to provide some interesting results which deserve more research.
2. The gloss relations seems to provide useful knowledge for propagating domain information through WordNet.

Obviously, some improvements and further investigation are needed with these new resources. For instance, we need to develop an automatic method to select which label or labels finally assign to a particular synset. Moreover, not all domains affect in the same way due to its initial distribution through the WordNet structure. We also need to investigate different combinations of relations for creating the knowledge base used by UKB. For instance, using only gloss relations, or a particular subset of WordNet relations.

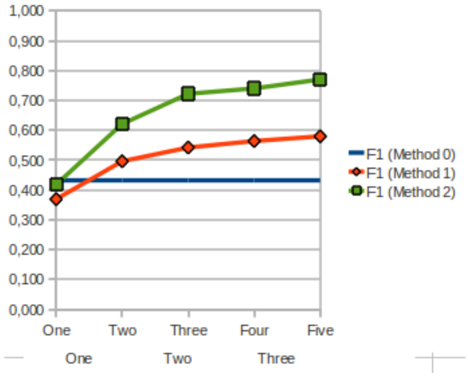


**Fig. 6.** Graphic showing *F1 score* values.

We also plan to try different combinations of methods and resources to improve the final result. For instance, we also plan to derive domain information from Wikipedia by exploiting WordNet++ [10].

## Acknowledgments

We thank the IXA NLP group from the Basque Country University. This work was been possible thanks to its support withing the framework of the KNOW2 (TIN2009-14715-C04-04) and PATHS (FP7-ICT-2009-6-270082) projects.

We also wish to thank the reviewers for their valuable comments.

## References

1. Vossen, P.: EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers (1998)
2. Magnini, B., Cavagli, G.: Integrating subject field codes into wordnet. In: Proceedings of the Second Internatgional Conference on Language Resources and Evaluation (LREC), Athens. Greece (2000)
3. Bentivogli, L., Forner, P., Magnini, B., Pianta, E.: Revising WordNet Domains hierarchy: Semantics, coverage, and balancing. In: Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources. (2004) 101–108
4. Magnini, B., Satrapparava, C., Pezzulo, G., Gliozzo, A.: The role of domains informations. In: In Word Sense Disambiguation, Treto, Cambridge (2002)
5. Fellbaum, C.: WordNet. An Electronic Lexical Database. Language, Speech, and Communication. The MIT Press (1998)
6. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009), Athens, Greece (2009)

7. Agirre, E., Cuadros, M., Rigau, G., Soroa, A.: Exploring knowledge bases for similarity. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10). European Language Resources Association (ELRA). Pages 373–377.". (2010)
8. Mihalcea, R., Moldovan, D.: eXtended WordNet: Progress Report. In: Proceedings of NAACL Workshop *WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, PA, USA (2001) 95–100
9. Castillo, M., Real, F., Asterias, J., Rigau, G.: The TALP systems for disambiguating WordNet glosses. In Mihalcea, R., Edmonds, P., eds.: Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, Association for Computational Linguistics (2004) 93–96
10. Navigli, R., Ponzetto, S.P.: Building a very large multilingual semantic network. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden (2010) 216–225