Extending the WN-Toolkit: dealing with polysemous words in the dictionary-based strategy

Antoni Oliver
Universitat Oberta de Catalunya (UOC)
Av. Tibidabo 39-43, 08035 Barcelona - Catalonia - (Spain)
aoliverg@uoc.edu

Abstract

In this paper we present an extension of the dictionary-based strategy for wordnet construction implemented in the WN-Toolkit. This strategy allows the extraction of information for polysemous English words if definitions and/or semantic relations are present in the dictionary. The WN-Toolkit is a freely available set of programs for the creation and expansion of wordnets using dictionary-based and parallel-corpus based strategies. In previous versions of the toolkit the dictionary-based strategy was only used for translating monosemous English variants. In the experiments we have used Omegawiki and Wiktionary and we present automatic evaluation results for 24 languages that have wordnets in the Open Multilingual Wordnet project. We have used these existing versions of the wordnet to perform an automatic evaluation.

1 Introduction

1.1 The WN-Toolkit

The WN-Toolkit[1] (Oliver, 2014) is a set of programs developed in Python for the automatic creation of wordnets following the expand model (Vossen, 1998), that is, by translation of the variants (words) associated with the Princeton WordNet synsets. The toolkit also provides some free language resources. These resources are preprocessed so they can be easily used with the toolkit.

The WN-Toolkit implements the following strategies for wordnet creation:

- Dictionary based methodology: This strategy uses bilingual dictionaries to translate the English variants associated with each synset. In previous versions of the toolkit this direct translation using dictionaries could be performed only on monosemic English, that is, variants associated to a single synset. About 82% of the English variants in the Princeton WordNet 3.0 are monosemic but frequent words tend to be polysemic. With the extension of the toolkit presented in this paper we are able to deal with polysemic English variants.

- Babelnet based strategies: BabelNet (Navigli and Ponzetto, 2010) is a semantic network and a multilingual encyclopedic dictionary with lexicographic and encyclopedic coverage of terms. In this methodology we simply extract the data from the BabelNet file to get the target wordnet. This strategy can only be applied to old versions of Babelnet, as new versions have a use restriction not allowing the creation of wordnets from its data.

- Parallel corpus based methodologies: In order to extract wordnets from a parallel corpus we need this parallel corpus to be semantically tagged with Princeton WordNet synsets in the English part. As these corpora are not easily available, we use two strategies for the automatic construction of the required corpora:

  - By machine translation of sense-tagged corpora.
  - By automatic sense-tagging of English-target language parallel corpora.

The WN-Toolkit also provides some resources, as dictionaries and preprocessed bilingual corpora.

[1] The WN-Toolkit can be freely downloaded from http://sourceforge.net/projects/wn-toolkit/

Language	Code	Synsets	Words	Senses	Core
Albanian	sqi	4,676	5,990	9,602	31%
Arabic	arb	10,165	14,595	21,751	48%
Basque	eus	29,413	26,240	48,934	71%
Bulgarian	bul	4,999	6,783	9,056	100%
Catalan	cat	45,826	46,531	70,622	81%
Chinese	cmn	42,312	61,533	79,809	100%
Croatian	hrv	23,122	29,010	47,906	100%
Danish	dan	4,476	4,468	5,859	81%
Finnish	fin	116,763	129,839	189,227	100%
French	fra	59,091	55,373	102,671	92%
Galician	glg	19,312	23,124	27,138	36%
Greek	ell	18,049	18,227	24,106	57%
Hebrew	heb	5,448	5,325	6,872	27%
Indonesian	ind	38,085	36,954	106,688	94%
Italian	ita	35,001	41,855	63,133	83%
Japanese	jpn	57,184	91,964	158,069	95%
Norwegian N.	nno	3,671	3,387	4,762	66%
Norwegian B.	nob	4,455	4,186	5,586	81%
Polish	pol	36,054	61,393	88,889	66%
Portuguese	por	43,895	54,071	74,012	84%
Slovene	slv	42,583	40,233	70,947	86%
Spanish	spa	38,512	36,681	57,764	76%
Swedish	swe	6,796	5,824	6,904	99%
Thai	tha	73,350	82,504	95,517	81%

Table 1: Statistics for the wordnets in OMW

1.2 The Open Multilingual Wordnet project

The Open Multilingual Wordnet[2] (OMW) (Bond and Paik, 2012) provides free access to several wordnets in a common format. We have performed experiments for 24 languages out of the 28 available wordnets. In table 1 we can observe some statistics about the wordnets for these languages. These wordnets have been used to perform an automatic evaluation of the results.

1.3 Omegawiki

Omegawiki[3] is a free collaborative dictionary that can be accessed through the Internet as well as downloaded as a relational database. The downloads are performed in MySQL dumps so it's easy to set up a MySQL database to have a local copy of Omegawiki. For our experiments we have downloaded all the sql dumps corresponding to the lexical data and we have created our own copy of Omegawiki. From this database we have extracted all the required data and we have filled up a new MySQL database according to the layout explained in section 2.1.

In table 2 we can observe the number of English-target language entries for Omegawiki for the languages in our experiments.

Omegawiki uses a complex set of semantic relations between its entries. It seems to be a great degree of freedom for the users to create new relations. A total number of 77 relations are found in the English Omegawiki, but only 22 of them has at least 50 occurrences. These relations can be observed in table 3.

We tried to relate the name of the relations in Omegawiki with standard relation names used in WordNet and Wiktionary (hypernym, hyponym, holonym, meronym, antonym and synonym). As holonym, meronym and antonym are already used in Omegawiki, we will try to find out the name used for hypernym, hyponym and synonym looking at examples of these relations in Wiktionary and observing if some of these examples are also present in Omegawiki. In this way we could establish the correspondence between relation codes and names in Omegawiki and standard relations names. An special case are synonyms, that are expressed as translations into the same language. In table 4 we can observe these correspondences.

In table 5 the number of definition and semantic relations in Omegawiki and Wiktionary can be observed.

[2] http://compling.hss.ntu.edu.sg/omw/

[3] http://www.omegawiki.org/

Language	Code	Omegawiki	Wiktionary
Albanian	sqi	417	4,431
Arabic	arb	3,293	17,157
Basque	eus	5,293	3,834
Bulgarian	bul	5,851	24,983
Catalan	cat	4,001	24,625
Chinese	cmn	3,368	70,553
Croatian	hrv	1,687	34,485*
Danish	dan	7,177	18,625
Finish	fin	9,654	94,193
French	fra	26,492	70,178
Galician	glg	1,636	7,832
Greek	ell	6,193	30,161
Hebrew	heb	3,447	12,452
Indonesian	ind	2,219	6,669
Italian	ita	25,083	51,098
Japanese	jpn	6,674	45,135
Norwegian N.	nno	787	5,842
Norwegian B.	nob	6,399	6,395
Polish	pol	8,280	32,486
Portuguese	por	11,858	58,925
Slovene	slv	5,102	9,036
Spanish	spa	36,139	63,512
Swedish	swe	10,271	45,016
Thai	tha	1,614	6,339

Table 2: Number of English-target language entries for each language

1.4 Wiktionary

Wiktionary[4] is also a free collaborative dictionary. This project is related with the Wikipedia and it is developed in a Mediawiki format. It can be accessed through the Internet and it can be also downloaded. The download format is an XML that includes sections in mediawiki format and for this reason it is difficult to parse.

The project Dbnary[5] (Sérasset, 2012) parses the Wiktionary content as soon as a new dump is available and provides this content in a easy to parse format.

In our first experiments we have used our own parser to extract the information for the English Wiktionary dumps but we missed a lot of information and it was very difficult and time consuming to correct the errors and expand the parser, so we started to use the results of the Dbanry project. We have used the files from Dbnary and we have stored all this information in our own database.

In table 2 we can observe the number of English-target language entries for Wiktionary for the languages in our experiments.

[4]urlhttps://www.wiktionary.org/

[5]urlhttp://kaiko.getalp.org/about-dbnary/

relation	freq.
is part of theme	16,158
parent	11,980
child	11,776
broader terms	7,299
narrower terms	5,639
is spoken in	4,692
related terms	3,717
borders on	797
is written in	633
antonym	328
official language	226
capital	209
country	192
wordt gevolgd door	178
currency	165
holonym	183
demonym	122
flows through	110
dialectal variant	78
meronym	73
flows into	68
is practiced by a	61

Table 3: Relations with at least 50 occurrences in English Omegawiki

Code OW	Relation OW	Relation S.
4	broader terms	hypernym
5	narrower terms	hyponyms
7574	antonym	antonym
375074	meronym	meronym
375078	holonym	holonym
-	translation into same language	synonym

Table 4: Conversion between Omegawiki (OW) relation codes and names and Standard (S.) relation names

2 Experimental results

2.1 MySQL database layout

We have stored all the data from Omegawiki and Wiktionary in our own MySQL database. This allows us to develop an algorithm for the construction of wordnets using this database and working in a independent way from the resource. This also allows us to add information from other sources and easily select one or more sources for the experiments. The database has the following 5 tables:

- *entry*: in this table the English word or expression, part of speech and source, along with an unique entry id are stored. The unique entry id allows us to select the information from the rest of the tables for a given entry.

- *translations*: in this table the translations for

	Omegawiki	Wiktionary
definitions	37,233	608,358
relations total	90,039	28,123
hypernyms	3,029	1,193
hyponyms	2,331	1,114
holonyms	121	92
meronyms	47	92
antonyms	171	0
synonyms	50,265	26,708

Table 5: Number of English definitions and semantic relations in the dictionaries

the target languages are stored, along with the language code and the entry id.

- *definition*: in this table the English definitions for each entry are stored.
- *tagged definition*: in order to avoid tagging each definition each time we perform an experiment we can use this table to store the tagged definition for each definition, along with the used tagger and the entry id.
- *relations*: in this table the related English words for each entry are stored along with the relation name and the entry id.

Please, note that most of the information we stored in the database is for the English language (except the translations). This is due to the fact that we plan to translate English variants from the Princeton WordNet in order to create the target wordnets.

Some indexes are create to speed up the algorithm. Most of the tables are converted into in-memory tables in order to further speed up the process of wordnet creation.

2.2 Algorithm

The wordnet extraction algorithm works as follows:

- Select all entry ids and target language words for a given target language and a given resource from the table *translations*.
  - For each entry id select the English words an pos from the table *entry*.
    - For the given English word and pos we search in the Princeton WordNet for all the synsets the word belongs to.
    - If the word belongs to one synset that means that it is monosemic and the target language word can be directly related to the given synset.
    - Otherwise, that means that it is polysemic and the disambiguation procedure is started:
      - Select all related words (hyponyms, hypernyms, holonyms, meronyms, antonyms and synonyms) along with the relation names from the table *relations*.
      - Select all the related words for all the synsets from the Princeton WordNet.
      - For each synset we count the coincident related words for each relation. A specific weight is given for each relation type.
      - Select the tagged definition from the table *tagged_definition* both for the definition coming from the dictionary as well as the Princeton WordNet definition. For each synset the coincident open class lemmata are counted and and specific weight is applied.
      - The synset with the higher score of weighted coincident relations and common open class lemmata in the definitions is attached to the target language word.

As we can see in the algorithm a set of weights has to be defined: a weight for each coincident type of relation and a weight for the number of coincident open class lemmata in the definitions. In our experiments a value of 5 has been used for all relations and a weight of 1 for coincident open class lemmata in the definitions. In section 2.4.4 a procedure for the optimization of the weights is presented.

2.3 Automatic evaluation procedure

We have used the existing wordnets in Open Multilingual Wordnet (OMW) for the 24 languages to perform an automatic evaluation. The evaluation procedure is as follows:

- Our algorithm gives us a set of synset-target language variants (SV) pairs.
  - If the extracted SV pair is also in the reference OMW, the result is evaluated as correct.
  - If the extracted SV pair is not in the reference OMW and there is other variants for the given synset in the reference OMW, the result is evaluated as incorrect.

– If the extracted SV pair is not in the reference OMW and there is no variants for the given synset in the reference OMW, the result is not evaluated.

The precision values obtained this way tend to be lower than the real values because the fact that some SV pair is not in the reference wordnet, but other variants for the same synset exist, doesn't really mean that the extracted SV pair is incorrect. May be is a valid variant for the synset, but this variant is not present in the reference wordnet.

2.4 Results

In table 6 we can observe the number of entries evaluated as correct, as incorrect and the number of entries that could not be evaluated since there is no information in the reference wordnet. The number of non-evaluated entries can give us an idea of the number of new entries we can add to the wordnet if a manual revision is performed

Lang.	Omegawiki			Wiktionary		
	C	I	N	C	I	N
sqi	58	45	249	296	207	2,263
arb	68	902	1,507	289	2,561	6,128
eus	1,339	694	881	1,192	532	635
bul	516	256	2,688	866	1,680	10,514
cat	1,671	680	554	5,697	2,915	3,881
cmn	857	526	1,344	3,640	8,140	14,775
hrv	785	274	287	2,151	7,120	4,757
dan	535	269	3,612	964	757	774
fin	3,778	2,309	18	17,551	21,325	127
fra	7,440	5,168	1,963	16,545	9,713	5,110
glg	589	134	561	1,579	498	2,328
ell	1,041	948	1,852	2,697	2,863	9,606
heb	29	575	2,018	133	1,390	5142
ind	919	484	259	1,704	1,383	758
ita	5,627	3,814	4,471	8,671	6,375	7,836
jpn	2,871	1,306	650	9,786	8,374	3,792
nno	70	17	517	326	222	2,668
nob	480	242	3,063	394	277	2,844
pol	2,348	1,310	1,434	6,133	4,402	5,817
por	4,832	1,810	474	12,892	7,741	5,410
slv	1,663	888	445	2,566	1,790	638
spa	4,088	4,567	8,525	6,179	7,155	15,274
swe	1,104	699	4,640	2,238	2,437	16,007
tha	733	464	85	1,639	1,632	330

Table 6: Figures of correct (C), incorrect (I) and nonevaluated (N) entries

In tables 7 and 8 the evaluation results are presented. For all the languages the number of extracted entries (synset-variant pairs) and the precision values (calculated in an automatic way) are presented, for several cases:

- *All no dis.*: All results, no disambiguation procedure performed.
- *All dis.*: All results, disambiguation procedure performed.
- *Non ambiguous*: Results corresponding to monosemous English variants (non ambiguous).
- *Amb. no dis.*: Results corresponding to polysemous English variants (ambiguous), no disambiguation procedure performed.
- *Amb. dis.*: Results corresponding to polysemous English variants (ambiguous), disambiguation procedure performed.

The comparison between the values with and without disambiguation procedure is interesting to observe the effectiveness of the disambiguation procedure. The results corresponding to monosemous English variants are interesting because they are the same we would obtain with the old version of the WN-Toolkit, that was not able to perform any disambiguation and was used only for monosemous English variants. They are also interested to be compared with the disambiguated results, to see if the figures are comparable.

In the tables some very low values of precision are present for languages as Arabic and Hebrew. They are due to languages specific features (as for example the writing of vowel signs than can be present or not both in the extracted variants and in the reference wordnet) that we were not able to cope with due to the our lack of knowledge of these languages. Other language-specific issues of the results will be explained in the section 2.4.3.

2.4.1 Results for Omegawiki

If we take a look at table 7 we can observe than the best overall results are obtained for Galician (precision of 81.47%) followed by Norwegian (Norsk) (precision of 80.46%). We must keep in mind that these values of precision are automatically calculated and the real values might be higher. If we concentrate on Galician we can observe than the precision of all results with no disambiguation procedure is 65.34%, so the disambiguation procedure improves the precision in 16.13 points. The precision for variants coming from monosemous English words is 83.43%, about 3 points higher than the overall values. If we concentrate on the variants coming from polysemous English words, we can see that the precision with no disambiguation is 51.16%, and it rises up to 76.85% (25.69 points) using the disambiguation algorithm.

2.4.2 Results for Wiktionary

If we now take a look at the results for Wiktionary at table 8 we can see that again the best results are

Lang.	All no dis.		All dis.		Non ambiguous		Amb. no dis.		Amb. dis.	
	Entries	Precision	Entries	Precision	Entries	Precision	Entries	Precision	Entries	Precision
sqi	1,466	40.18	353	56.31	135	58.33	1,332	38.76	219	55.7
arb	9,191	4.19	2,478	7.01	1,237	9.83	7,955	3.34	1,242	4.97
eus	7,934	48.03	2,915	65.86	1,708	64.99	6,227	43.77	1,208	66.8
bul	29,183	36.66	3,461	66.84	1,862	63.09	27,322	35.66	1,600	68.46
cat	8,531	53.57	2,906	71.08	1,673	69.76	6,859	48.74	1,234	72.81
cmn-Hans	11,924	26.88	2,728	61.97	1,269	68.88	10,656	22.39	1,460	57.71
hrv	4,180	51.59	1,347	74.13	701	78.89	3,480	43.18	647	68.4
dan	11,935	48.38	4,417	66.54	2,523	58.3	9,413	46.8	1,895	70.73
fin	20,134	36.02	6,106	62.07	3,342	64.24	16,793	30.4	2,765	59.44
fra	53,499	48.3	14,572	59.01	7,850	57.48	45,650	46.75	6,723	60.74
glg	3,483	65.34	1,285	81.47	753	83.43	2,731	51.16	533	76.85
ell	12,838	34.61	3,842	52.34	2,009	52.29	10,830	31.09	1,834	52.38
heb	9,199	2.56	2,623	4.8	1,347	4.37	7,853	2.22	1,277	5.11
ind	5,589	48.06	1,663	65.5	852	64.68	4,738	44.86	812	66.33
ita	85,324	32.05	13,913	59.6	6,614	59.82	78,711	29.41	7,300	59.41
jpn	14,994	40.48	4,828	68.73	2,694	71.59	12,301	33.16	2,135	65.32
nno	1,379	59.89	605	80.46	376	80.0	1,004	55.92	230	80.7
nob	10,555	47.0	3,786	66.48	2,196	58.61	8,360	45.21	1,591	70.5
pol	16,417	41.99	5,093	64.19	2,876	64.67	13,542	35.37	2,218	63.55
por	26,301	52.52	7,117	72.75	3,761	69.16	22,541	48.36	3,357	77.10
slv	9,136	49.68	2,997	65.19	1,607	61.59	7,530	47.1	1,391	69.21
spa	68,884	31.65	17,181	47.23	8,874	41.86	60,011	30.55	8,308	51.41
swe	21,626	40.05	6,444	61.23	3,535	63.17	18,092	35.67	2,910	59.81
tha	4,065	33.08	1,283	61.24	677	59.87	3,389	27.12	607	62.72

Table 7: Results for Omegawiki

obtained for Galician (a precision of 76.02% for all the results with disambiguation). The rest of figures for this languages follows the same pattern as for Omegawiki. One important fact is that with Wiktionary we are obtaining much more results (4,406 synset-variant pairs) than with Omegawiki (1,285) as Wiktionary is a much bigger resource as can be observed in table 2

2.4.3 Comments on the results

The precision values for the experiments are very different for each languages. It can be due to several reasons, for example:

- The quality of the dictionary (Omegawiki and Wiktionary) for each language can be different, as they are collaborative dictionaries. Not only the size of the resource is important, but also the precision of the translations.
- The quality and completeness of the reference wordnet in OMW. Here again not only the size (number of synstet-variant pairs) but also the number of possible variants for the same synset are very important.

There are a lot of language-specific issues in the dictionaries and reference wordnets that must be taken into account. We already mentioned the writing of vowel signs in Arabic and Hebrew, that we could not cope with due to the lack of knowledge of these languages.

For example, if we observe the results for Bulgarian, we can see that precision for Omegawiki (66.84%) is much higher than precision for Wiktionary (34.01%). The main reason is that in Wiktionary most entries are marked with accents in vowels to express the stress (for example алкохол in Omegawiki but алкохóл in Wiktionary). This marks are not used in standard writing and so they are not used in the reference OMW wordnet. To use the Wiktionary results a simple script converting the accented characters to unaccented can be used.

For Croatian we face a double problem. Both in Omegawiki and Wiktionary some entries (but not all) are using the diacritics on vowels to express stress and intonation, but these symbols are not used in the reference OMW wordnet as they are not used in standard writing. This can also be solved with the use of a simple script. On the other hand, Wiktionary is not using a language code for Croatian (hrv) but one for Serbo-Croatian or Croatian-Bosnian-Serbian macrolanguage (hbs). Entries for this code can be Croatian words written in latin but also Serbian words written in cyrillic or latin. As in the Croatian reference OMW wordnet there are only standard Croatian

Lang.	All no dis.		All dis.		Non ambiguous		Amb. no dis.		Amb. dis.	
	Entries	**Precision**	**Entries**	**Precision**	**Entries**	**Precision**	**Entries**	**Precision**	**Entries**	**Precision**
sqi	11,510	43.02	2,767	58.85	1,251	59.03	10,260	41.91	1,517	58.77
arb	37,540	6.75	8,980	10.14	4,431	12.3	33,110	6.21	4,550	8.79
eus	9,359	50.7	2,360	69.14	1,244	69.89	8,116	47.41	1,117	68.43
bul	59,664	25.23	13,061	34.01	5,690	34.95	53,975	24.53	7,372	33.63
cat	53,737	52.1	12,494	66.15	6,597	68.86	47,141	49.5	5,898	63.22
cmn	102,130	18.98	26,519	31.47	30	36.36	88,589	16.79	14	25.0
hrv	62,765	17.4	14,029	23.2	6,399	25.57	56,367	16.17	7,631	20.99
dan	43,052	39.95	9,469	56.01	4,866	62.01	38,187	38.4	4,604	53.65
fin	174,743	26.22	39,004	45.15	19,958	54.95	154,786	22.51	19,047	34.88
fra	119,160	53.91	31,369	63.01	17,802	66.24	101,359	51.67	13,568	58.51
glg	17,745	59.92	4,406	76.02	2,261	77.95	15,485	52.99	2,146	72.66
ell	67,014	32.3	15,168	48.51	7,408	55.4	59,607	29.85	7,761	43.35
heb	32,136	4.97	6,666	8.73	3,198	10.31	28,939	4.18	3,469	7.33
ind	17,341	41.1	3,846	55.2	1,799	54.55	15,543	39.56	2,048	55.74
ita	95,540	39.5	22,883	57.63	12,093	64.59	83,448	35.39	10,791	50.04
jpn	89,706	31.92	21,954	53.89	11,423	63.19	78,284	27.08	10,532	43.7
nno	11,670	47.37	3,217	59.49	1,751	64.94	9,920	45.66	1,467	56.95
nob	13,012	47.01	3,516	58.72	1,855	63.13	11,158	45.42	1,662	56.61
pol	69,365	36.29	16,353	58.22	8,398	65.55	60,968	30.91	7,956	49.82
por	120,069	46.11	26,044	62.48	13,486	65.64	106,584	42.82	12,559	58.84
slv	25,391	47.17	4,995	58.91	2,248	59.59	23,144	45.91	2,748	58.33
spa	114,452	38.68	28,609	46.34	15,517	46.46	98,936	37.78	13,093	46.22
swe	93,448	32.08	20,683	47.87	10,637	57.12	82,812	29.12	10,047	40.69
tha	15,660	27.77	3,602	50.11	1,784	53.62	13,877	23.85	1,819	46.56

Table 8: Results for Wiktionary

words, the values of precision for Wiktionary are lower.

So it is important that a native speaker of each language revise the obtained results in order to detect these issues and try to solve them in an automatic way.

2.4.4 Optimization of the weights

In the experiments we have used a fixed value for the weight for the different relations and common lemmata in the definitions. The extraction algorithm can give also a file with information about all the parameters. Here we can see an example for Catalan:

```
pluja àcida MONO 14517629-n
àcid POLY 14607521-n/2:1:0:0:0:0:0;
02675657-n/0:0:0:0:0:0:0
```

The first line tell us than *pluja àcida* comes from a monosemic English word having the synset 14517629-n. In the second line we can learn that *àcid* comes from a polysemic English word that is a valid variant for the synsets 14607521-n and 02675657-n. For the first synset we have two common lemmata in the definitions and one common hyponym, whereas for the second synset we don't have any information in common.

This file allow us to experiment with different weights in order to learn the best combination. In table 9 we can observe the values of overall precision for different combinations of the parameters (we have assigned one weight to the coincident lemmata in the definition and another weight for the coincident related words (the same weight for all types of relations). The values in the table are for Catalan and for Omegawiki and Wiktionary.

Def.	Rel.	Omegawiki	Wiktionary
0	1	70.01	**68.90**
1	0	70.95	66.15
1	1	71.03	66.14
1	5	**71.08**	66.15
5	1	70.98	66.14
1	10	71.06	65.90
10	1	70.98	66.14

Table 9: Precision for different combinations of the weights for Catalan

As we can observe, the best combination for Omegawiki is 1 for definition and 5 for relations. This is the combination we have used in our experiments. For Wiktionary the best combination is 0-1, that is, using no definitions and using only relations.

It would be worth to do a better analysis and to try to use some machine learning technique to find the best combination for each languages and resource.

3 Conclusions and future work

In this paper we have presented an extension of the WN-Toolkit that allows to use the dictionary-based technique for wordnet creation for English polysemous variants, provided that the dictionary has definitions and/or relations. The algorithm have been applied to 24 languages having wordnets available in the Open Multilingual Wordnet. We have calculated values of precision in an automatic way using as reference the existing wordnets. For the experiments we have used two freely available dictionaries: Omegawiki and Wiktionary. The results demonstrate that the algorithm performs well in the task of selecting the correct translation for polysemous words.

As a future work we plan to use some machine learning technique to try to find the best combination of parameters for each language and resource. The algorithm we've presented uses a very simple strategy to find the most similar definition by comparing the number of coincident open class words. We plan to experiment with more complex strategies, as for example using a word2vec approach or similar techniques (Bjerva et al., 2014). We also plan to use other dictionaries or encyclopedia as Apertium transfer dictionaries, Wikispecies, Wikipedia, Geodata, as well as proprietary dictionaries under agreement with the copyright holders. If the dictionary has definitions and/or semantic relations the proposed disambiguation algorithm can be applied. If not, only target language variants corresponding to English monosemous variants can be extracted.

We also plan to run the algorithm for all languages in the resources, creating preliminary wordnet versions for languages not having freely available wordnet available. In this sense we would be happy to make agreement with universities or institutions in target language speaking countries to revise the results.

We want to compare and share the results with the Extended Open Multilingual Wordnet (Bond and Foster, 2013).

An lastly we want to pack the new algorithm into the WN-Toolkit and share the complete MySQL database created from the free resources. This database can be useful for wordnet creation experiment as well as for other lexicographical tasks.

Acknowledgments

This research has been partially carried out thanks to the Project SKATER (TIN2012-38584-C06-01 and TIN2012-38584-C06-06) supported by the Ministry of Economy and Competitiveness of the Spanish Government.

This research has been done during a research stay in the Vrije Universiteit in Amsterdam, thanks to a mobility grant from the Universitat Oberta de Catalunya. I would also thank Piek Vossen and his research group for welcoming me in Amsterdam.

References

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646, Dublin, Ireland.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*, Sofia, Bulgaria. 1352–1362.

Francis Bond and Kyonghee Paik. 2012. A survey of wordnets and their licenses. In *Proceedings of the 6th Global WordNet Conference (GWC 2012)*, Matsue, Japan. 64–71.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Stroudsburg, PA, USA. Association for Computational Linguistics. ACM ID: 1858704.

Antoni Oliver. 2014. WN-Toolkit: Automatic generation of WordNets following the expand model. In Heili Orav, Christianne Fellbaum, and Piek Vossen, editors, *Proceedings of the 7th Global WordNet Conference*, pages 7–15, Tartu, Estonia. Global Wordnet Association.

Gilles Sérasset. 2012. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web Journal-Special issue on Multilingual Linked Open Data*.

Piek Vossen. 1998. Introduction to eurowordnet. In Piek Vossen, editor, *EuroWordNet: A multilingual database with lexical semantic networks*, pages 1–17. Springer Netherlands.